RESEARCH THROUGH DESIGN OF BENDABLE INTERACTIVE PLAYING CARDS

A DISSERTATION SUBMITTED TO THE GRADUATE DIVISION OF THE
UNIVERSITY OF HAWAIʻI AT MĀNOA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

COMPUTER SCIENCE

DECEMBER 2021

By

Nurit Kirshenbaum

Dissertation Committee:

Scott Robertson, Chairperson
Jason Leigh
Martha Crosby
Kimberly Binsted
Monique Chyba

Keywords: Human Computer Interaction, Tangible User Interface, Bendable Devices, Research through Design

ACKNOWLEDGMENTS

This... wasn't easy.

However, I'm grateful for the opportunity to go through with it, and for all the things I learned along the way. I wasn't always sure what to do next. Actually, I was never sure. But now, looking back, I feel a sense of accomplishment that will surely last for at least a week. I am excited to move on towards the next part of my life.

I would like to thank my committee members: Scott Robertson, Jason Leigh (my fairy godmother), Martha Crosby, Kimberly Binsted, and Monique Chyba. Scott, thank you for pulling me up when I was feeling down. I also like to thank the various other faculty and staff at the Information and Computer Science department at the University of Hawaii at Manoa who taught and supported me: Dan Suthers, Edo Biagioni, Lipyeow Lim, Guylaine Poisson, Kyungim Baek, and Janice Oda-Ng – Mahalo!

I am grateful for all the great people I got to work with in the Laboratory for Advanced Visualizations and Application (LAVA). Thank you Ryan Theriot, Roderick Tabalba, Justin Bishay, Katrina Turner, Alberto Gonzalez Martinez, Troy Wooton, James Hutchison and everyone else. Special thanks to Dylan Kobayashi, Kari Noe, and Eva Morales Perez who contributed to my dissertation work.

Some thank you wishes sent to the other side of the world: thank you Sara and Avi Kirshenbaum (also known as mom and dad) and my sisters and their families, the Levi family, Hila, Yaron, Dror, Alon, and the Mendalawi family, Keren, Moshe, Maor, Yoav, and Yotam. I miss you all daily and hope you are proud of me. Thank you to Galya Pasi Kitaro and Ziv Kitaro, who, whether they'd like to or not, are by best friends.

And Finally - thank you Asaf for everything that you are and everything that you do for me, the support and the cooking and the laughing and the listening to my frustrated complaints even when you have no idea what I'm talking about. And thank you my boys, Shoni and Ethan. I love you. You are my everything. I would try to say something funny, but you would probably say that it's lame.

ABSTRACT

Computer Interaction has become second nature for almost all modern people and touch interaction on smart devices has likewise become ubiquitous. It is easy to forget how new the touch interaction paradigms are and the path it took to develop them. Nowadays, interaction designers are looking for even more novel interaction techniques with previously unheard of input and output channels. One direction for this search is bendable interfaces - interfaces that require their users to bend them as a form of interaction. In this work, I will overview and analyze a collection of prior academic research relating to bendable devices. Researchers often wonder: what will work well with bend interactions? In this dissertation I offer the answer "bendable interactive playing cards", and I frame my work on this word-salad using the Research through Design methodology.

Ultimately, I hope to answer the question: Is bending interaction suitable, feasible, and expressive for interactive playing cards?

My interactive playing card devices, which I call PEPA (Paper-like Entertainment Platform Agents) are inspired by my love of both paper-based and digital card games. By combining computational capabilities in multiple stand-alone physical devices, I can offer more than the two media forms can offer separately. I describe 6 possible scenarios where such a system can be used as well as other hybrid digital-physical game systems inspired by card and board games. Of course, the concept of interactive playing cards does not automatically lend itself to bend interaction, so I will try to justify this integration of ideas via a study of the literature and my observations of card players.

Following my arguments to incorporate bending and interactive cards, I created a proof-of-concept prototype. In true Research through Design form, this was a situation where one has to build an object before they can understand what research directions to take. In this case, the prototype led to further user studies regarding the timing of actions during the bend gesture and a model for bend events. At a different point, I used design as a research activity when I conducted a workshop for designing games for interactive cards. I will report the procedure, results and analysis from this workshop to illustrate the design space of possible games.

Research through Design is a research approach within the field of HCI that has multiple, sometimes conflicting, interpretations. It is mostly agreed that such research involves the creation of some prototype and an end goal of extracting and disseminating knowledge. In this work I will present the different approaches for documenting RtD as well as my own contribution: the Designer's Reframe and Refine Diagram. This is a method that uses a diagram as a tool to reflect on the design process as a whole in a prototype-centric way. I will show how I use this method to systematically document 5 versions of prototype in the PEPA project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

The idea of using bending as an interaction gesture goes back to the 90s, when interaction designers were looking for ways to break away from the successful WIMP (Windows, Icons, Mouse, Pointer) paradigm. The Post-WIMP interaction movement spawned many modern devices that use tangible interaction (such as mobile tablets and smartphones), gesture recognition (such as Kinect or Leap Motion), and voice commands (such as personal and home assistants like Siri or Alexa). However, despite early prototypes like the DataGlove and ShapeTape, bending has not caught-on as a method of interaction.

A renewed interest in bending gestures blossomed with the release in 2004 of the work on Gummi. The paper presented both a vision for flexible displays and computers and a prototype for a device with a rigid display on a flexible substrate controlled via bending. In the following chapters, I will go more in-depth into Gummi and the works that followed it to expand our body of knowledge regarding bendable devices. I divide these works to four categories with distinct purpose and characteristics: Foldables, Device Peripheries, Personal Device, and Digital Paper.

The Digital Paper category is of particular interest, since it tries to bring digital enhancement to a known metaphor - paper documents. Moreover, since natural use of paper commonly involves bending (along with other harsher gestures such as folding, crumpling, cutting etc.), this metaphor is a good fit for bend interactions. Unlike regular pieces of paper, the digital paper system can have dynamic content that changes based on context and needs. Having multiple units of digital paper affords a user sophisticated digital capabilities along with the tactile feedback and 3D spatial organization of paper.

While other researchers focus on digital paper as an advanced system for working with documents, I will introduce in this work an adjacent metaphor from a slightly different domain, namely, playing cards. Playing cards have been in use for centuries and across many cultures and are a widely-known form of entertainment. In recent years, as a strong community of boardgames hobbyists and enthusiasts evolved, there has been an increase in complex card games (some belonging to the Collectable Card Games, CCG, category) such as Pokemon, Magic: The Gathering, and Munchkin. This trend proceeded into the digital realm with top ranking card game apps such as Hearthstone, Shadowverse, and Chrono magia. In this dissertation, I will present an observational study in which pairs of experienced players played paper card games in a recorded lab environment, and the ensuing analysis, which indicates that games are rhythmic and incidental bends co-occur with key game actions. My analysis will show that bend gestures are a promising interaction scheme for digital cards.

However, bend gestures are not simple to interpret - a factor that may have had an effect on the difficulty to find suitable applications for bending so far. For one thing, the degrees of freedom involved in a bend can be limited (like the devices presented in this work) or freeform, limited only by substrate connectivity. The fully freeform bendable devices have complexity levels similar to full-body gesture recognition systems. Another contention point for interpretation is the possible dual nature of bends as either discrete (causing changes upon reaching specific angles of bend) or continuous (fluidly causing change while the angle of a bend changes). Researchers used different approaches in their implementations, variating on discrete and continuous events, and occasionally using additional input methods (such as buttons) to complement the bends.

If we stop and think about even the simplest of bends - bending around a single axis either vertically or horizontally - there are many phases along the way. When you start, the surface is flat, in neutral position, then you start applying pressure to create a bend and continue changing the angle you are forming; next you reach a high point of the bend, so you start releasing the pressure you applied, once again changing the angle you are forming, but in the opposite direction, until the surface is flat again. The authors of the visionary Gummi paper, realizing the complexity of the gesture, tried to define bending states and events that would be used to describe it. However, subsequent work showed that there are other plausible event schemes. The event model I will present in this dissertation is a fusion of current approaches to bend events and a synthesis of the results from a user study in which participants had to decide the specific point in the bend gesture they would expect to see a change in the device's output.

Designing bendable interactive cards, like any other design task, is a "wicked" problem with unclear formulation of requirements. The design includes 1) technical aspects of making the cards function as an interactive device, 2) aesthetic issues that affect the device's affordance as a card, and 3) decisions about materials that would facilitate bending. In this dissertation, I will fully detail the process of prototype design over the last few years. First, I will introduce my Designer's Reframe and Refine Diagram, which is a visualization tool and method I formulated to help me systematically document my Research through Design process in a way that corresponds to the way I was thinking about the project. Then, I will detail five reframe and refine stops in my prototyping journey, from proof-of-concept prototype, to a slick design that is easy to fabricate. I named the devices I created as part of this project **Paper-like Entertainment Platform Agents** or PEPA for short.

With the concept of bendable card games at hand, the bend events model formed, and the card device implemented, I will end my work as part of this dissertation with a workshop study. In this workshop, I asked designers to come up with game ideas for interactive cards and for bendable interactive cards in a brainstorming session. I will present the game ideas my fellow designers generated, along with a Qualitative Content Analysis I preformed on these ideas. I will draw conclusions regarding the potential versatility of the design space for interactive cards.

Finally, I close this manuscript with an agenda for future research directions.

1.1 Why Playing Cards?

Card games are ubiquitous. The variety of games available to card players has been steadily increasing in the last few decades as the gaming industry grew overall. Trading Card Games (TCG) have especially gained popularity with ongoing best-selling games such as *Magic: the Gathering* [125] (abbreviated M:TG), and *Pokemon*. TCG games offer multiple layers of enjoyment - like any other card game players enjoy facing opponents in matches, but they also enjoy the collecting aspect of the game, and the deck building aspect of the game where they select a limited amount of cards strategically in hopes of defeating their opponents' deck.

In addition to the ubiquity of paper-based card games for both casual and enthusiast players, card games make for popular video games as well. The digital card games market has proven to be lucrative, earning $1.4 billion in 2017 [168]. In the lead is *Hearthstone* (see figure 1.1c) [15], a digital TCG published in 2014, which has been a guiding beacon to other digital card games ever since.

There are advantages to both paper-based card games, and digital ones. The paper cards are usually played face to face as a group of players, encouraging socialization and communication. Digital card games, however, introduce innovative game mechanics that rely on computerized capabilities to enhance the game,

Figure 1.1: Three realizations of card games

such as animation, timed events, and game state sensitive rules.

I envision a third realization of card play, *interactive cards*, which harness both the tactile and social characteristics of paper cards as well as computer supported game mechanics. With the advancement of thinner and flexible electrical components and displays, card devices that posses tangible features like paper cards and behave like digital card games, seem viable in the near future. In this dissertation, I will suggest bending as a form of interaction for such cards.

1.2 Scenarios

In this section, I detail possible usage scenarios of interactive cards. These are based on both related work and original ideas.

Automate Card Play

The strength of interactive cards where each card has a digital displays is the flexibility to change systematically what a card shows. This can be used to automate several common activities that players perform while playing paper-based card games. *Shuffling* cards can be efficiently done by a computer, instantaneously causing a new arrangement of virtual cards on the interactive cards. Many games call for *secondary para-*

phernalia during card play, such as dice, life points, and tokens that modify card information, for example, the health of a creature shown on a card may decrease as it battles other creatures, and players may indicate new values with tokens. The interactive card system can also allow easy *deck selection* and switching from one deck to another without the need to carry multiple decks. It should be noted that TCG players enjoy the card collecting aspect of the game and may not find it disadvantageous to carry superfluous cards, but new platforms sometimes change people's views.

> Zoe is a card collector. When she buys a new card, she scans it to her online collection of PEPA cards. This allows her to arrange many different decks and switch between them with a few clicks on her tablet. With another click, her deck is shuffled and she can begin to play. When she plays a resource card, she can use it to update resource cards already in play. When she plays a creature card, bending the creature and resource cards together updates the resources according to the creature's cost. When she attacks her opponent, she bends her creature and the opponent's creature cards together to update the new health values. All along, the general state of the game for both players is automatically updated on the tablet.

Enrich Game Experience

One familiar property of digital formats that paper based cards cannot support is the incorporation of rich media, animations and sounds. This form of experience enhancement has been the inspiration for some of the related works using AR and paper cards to create an experience that is both tactile and immersive. The interactive cards offer a similar tactile advantage while supporting animations on each individual card, regardless of its position, as well as animations and sounds on the accompanying mobile device.

> Ariel's deck has cards for a Wizard, Acolytes, and Treants characters in a card game loaded to his PEPA deck. Each character has a simple avatar and idle animation that change depending on the spells and artifacts applied to them. When Ariel bends an acolyte and a spell book cards together, an audio story line is played accompanied by an animation. In addition to visually helping keep track of all the character's modifications, Ariel feels that the game experience is more fun and immersive.

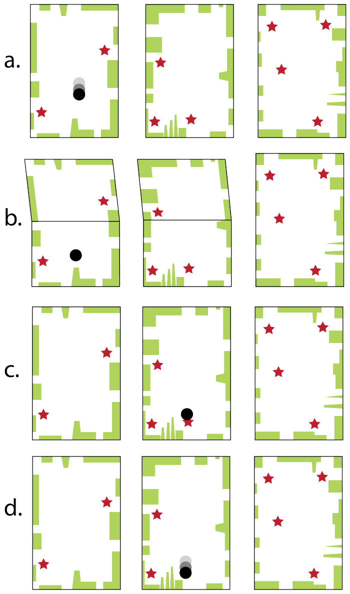


Figure 1.2: 3 card pinball (concept illustration)

Guided Games

Learning a new game can be hard and novice players may need some time before they get used to new rules and remember which cards are playable at what part of the game, and even more time before they can develop strategies to select to best play. Since the interactive cards are a digital platform, a player can specify that they want to play in a guided mode. In this mode, they may receive current instructions that pertain to the phase of the game on their device. Each card may have additional information to indicate if it is playable at the time. For example, the digital game Hearthstone highlights with a glowing frame playable cards. Additional markings can rate how advantageous the possible playable cards are.

> Kacey is learning a new card game. Her friend, Micah, gave her a general explanation of the rules, but she still feels unsure. She activates the tutorial mode in her PEPA deck. A brief outline of each round is presented on the tablet. Her cards show an additional small markers for cards she may play according to the resources and other cards she has in play. Bending the card away from her will give her more detailed information about it, such as, what other cards can it attack. When she makes an attack, bending her and her opponent's cards, her card warns her if the action is not legal and for what reason. Her friend still helps

> her with these initial rounds, however, she feels more agency to play the game without fumbling on basic rules.

Accessibility is Playing Cards

Making each card interactive and incorporating a connection to a mobile device, opens new opportunities to make card games more accessible. For example, visually impaired players can gain access to all card games on this platform. When a player wants to know the details of one of their cards or their opponent's cards, they can interact with that card - for example, bending it - invoking an audio message with the desired information from the mobile device. A more elaborate form of interaction can be devised, such as bending the card while touching the mobile device, if the game itself is interactive. The player can use earphones to prevent their opponent from hearing which cards they are holding.

> Kai, who is visually impaired, enjoys playing cards with his family and friends. Using the PEPA system, he places an earphone connected to the tablet in one ear, and draws his initial hand. Bending a card backward will prompt the system to read the card description and statistics. Bending his and his opponent's cards that are already in play on the board reads an updated description of the cards and their status. When his opponent makes a play, the system automatically provides an audio description of the play, helping Kai maintain the game state.

Novel Game Mechanics

The proposed platform introduces a new way of play and interaction. When the Wii platform was released with its new form of interaction, new kinds of games were designed to take advantage of it, and we expect a similar opportunity with the interactive cards. They are not restricted to static cards that are embellished, but can inspire completely new game mechanics that are not currently used in paper based games or digital card games.

> Laura is using the PEPA deck to play three card pinball (Figure 1.2). Laura bends pairs of cards to make the ball switch between them, collecting stars and avoiding obstacles. When

> a card has lost all of its points, she can bend it backwards to shuffle a new layout onto it. When her friend Miguel decides to join the game, he picks up 3 cards of his own and activates them by bending them backwards. Now, he can steal the ball from her, until she steals it back, and mayhem ensues.

Hybrid Interactive Cards and Remote Play

While we stress throughout this paper the importance of the social component in card playing, we cannot disregard the aptitude of a computerized system, especially one that utilizes a mobile device, to facilitate remote play. The player can use their own interactive cards on their side of the game and the state of the game will be sent to the opponent who may also play with interactive cards on their side. The players will be able to see each other's board on the mobile device, and the interactions required to play may involve both the physical cards and the touch screen of the mobile device. Likewise, one of the players may use the interactive cards while the other uses a fully digital application on their mobile device.

> Koby uses the mobile application to play against online community members. He can choose if he wants to use his physical interactive cards, or their digital avatars. While using his PEPA deck, when he plays a card by bending it, the card appears on his tableau on the application. He can use touch and bend combinations to replicate actions that would usually require bending two cards.

1.3 Research Questions

The overarching question guiding this dissertation is: **Is bending interaction suitable, feasible, and expressive for interactive playing cards?** This question has three components 1) suitability, 2) feasibility, and 3) expressivity.

To address suitability, I first conducted a meta-analysis of related works on bendable devices, and evaluate the idea of interactive playing cards as it relates to existing work.

RQ 1- Is bending interaction appropriate for interactive playing cards based on prior work?

Then, I conducted an observational study, analyzing the hand movements of 4 players engaged in card games, and evaluated the likelihood of bending to appear as a natural gesture.

RQ 2 - How are cards bent in a real-world game scenario?

To address feasibility, I explore what it takes to make the hardware system of a card, and how to make the software governing the system easy to use from an application developer point of view. I will describe the design iterations of the card prototypes.

RQ 3 - How to make a bendable interactive card?

I will also describe a study into an event model that facilitates app development.

RQ 4 - What is the mental model of users bending a device?

RQ 5 - How to define bend events?

To address expressivity, I preform a study that includes a design workshop for PEPA games and analyze the resulting game pitches for features of expressivity.

RQ 6 - Are bendable interactive cards expressive?

1.4 Outline

Chapter 1 - An introduction to the project "Research through Design of Bendable Interactive Playing Cards" and PEPA devices, research questions, motivation, and scenarios.

Chapter 2 - A literature review of several topics with relevance to the content of this work: Tangible User Interfaces (TUI), Input devices and techniques (particularly events), Digital/Physical hybrid boardgames with attention to digitizing non-digital games and the cross between hybrid boardgames and card games, and Research through Design (RtD).

Chapter 3 - Meta-Analysis of bendable devices: in depth review and categorization of prior work with attention to categories, applications, gestures, and research activity.

Chapter 4 - A study of naturalistic gestures in paper-card play, an observation of players interacting with cards during play.

Chapter 5 - A user study of perceived expected effect timing when bending a card in different contexts. A mental model of bend events and a practical model of bend events to accommodate potential problems in natural bend signals.

Chapter 6 - A description of the Designer's Reframe and Refine Diagram (DRRD) method and tool. DRRD applied to the PEPA project: description, goals, refinements, and lessons learned applied to 5 prototype versions.

Chapter 7 - A workshop study in which designers brainstormed ideas for interactive card games. The generated ideas were analyzed using a Qualitative Content Analysis method.

Chapter 8 - Conclusion and 6 future research directions for bendable card games.

1.5 Contributions

- Meta-analysis of bendable devices in research literature: what do we know about bendable devices?
- Empirical observations that grounds bend interactions as natural gestures while playing cards.
- A framework for handling bend events: a description of an event model motivated by empirical observations.
- Interactive Card System implementation: the creation process of bendable cards and the software that drives them.
- Designer's Reframe and Refine Diagram (DRRD): a tool for reflection in the design process of Research through Design from a prototype-centric point of view.
- A use case of (DRRD): methodically applying DRRD to the bendable interactive cards project.
- A workshop study exploring the design space of games for interactive cards.
- Future research agenda into interactive cards and bendable devices.

CHAPTER 2
BACKGROUND

2.1 Tangible, Organic and Bendable User Interfaces

At the 1990 SIGGRAPH conference, Mark Green and Robert Jacob conducted a workshop discussing Post-WIMP (called Non-WIMP at the time) user interfaces [59]. The goal of the workshop was to remind the User Interface research community that despite the success of the desktop metaphor during the 80's and the WIMP (Windows, Icons, Mouse, Pointer) paradigm that had become common, there is a need to continue research on Non-WIMP interfaces. These included virtual reality interfaces, devices with embedded electronics, gesture based interaction and other modalities. Decades later, we enjoy the fruits of the research highlighted in that workshop with commercially successful Post-WIMP devices such as mobile personal devices, smart watches, VR sets, and voice assistant devices. Already, several models of devices with foldable screens are available for purchase, which touches on a category of devices covered by the bendable umbrella.

The Post-WIMP novel interaction paradigm covered here involves the use of physical objects within digital systems. Ishii and Ullmer coined the term **Tangible User Interfaces** (then - Tangible Bits, now known as TUI) for computationally interpreted objects. They say *"Tangible Bits allows users to 'grasp & manipulate' bits in the center of users' attention by coupling the bits with everyday physical objects and architectural surfaces."* [80] However, the exact definition and extent of TUIs remained vague - for example, keyboards are tangible objets that help manipulate digital 'bits' but are not considered a Post-WIMP, TUI device [42]. Others tried to find better ways to describe TUI systems [69, 42, 152].

There are countless examples of TUI systems developed in research environments; and these sometimes plant roots strong enough to become mainstream products. Siftables [119] is a system of block-like devices that communicate with each other created by Merrill for his dissertation as a game platform. The reacTable project [83, 82] made table-top use of tangible object accessible to many. SandScape and illuminating Clay [79, 77] introduced the concept of changing the shape of a material like sand to form new topographies, this is now seen in many museums. Some systems are not yet at product level, but are highly promising, such as inForm [104], an actuated display, or toy-like programming blocks [71, 170].

Many frameworks were developed to help the creation of TUI prototypes: Papier-Mache [97, 96], ARToolKit [73], Sketch-a-TUI and Paperbox [176, 175], Phidgets [60], and d.tools [64, 63]. However, these usually do not present examples that use bend sensors.

Tangible User Interfaces has grown wildly as a field [151] and even has a dedicated conference these days[1].

Organic User Interfaces (OUI) promote the vision that in the future the shape of displays will become a decidable variable during the design of technology. Displays may be able to take any arbitrary shape and the shape itself may be dynamic. Holman and Vertegaal [67] identify the origin of their use of the 'organic' descriptor from Organic Architecture, a concept coined by Frank Lloyd Wright in 1939 expressing a desire for balance between human and natural design. They explain that *"An Organic User Interface is a computer interface that uses a non-planar display as a primary means of output, as well as input. When flexible, OUIs have the ability to become the data on display through deformation, either via manipulation or actuation."*

Bendable devices, as an overarching class of devices that use bending as a form of input, have not been formally defined, so I present my own definition of bendable devices as devices in the intersection of TUI and OUI that can identify bending of the device's surface as input. A bendable device will have (1) a display (output) as part of the device, and (2) will use a grasp and bend motion to trigger an output change (bend events).

The use of bend sensor to enhance user experience is not new. The DataGlove [183], for example, allowed the direct manipulation of computer-generated objects by identifying hand gestures using optical bend sensors stretched across the joints of a glove. ShapeTape [7], as another example, used a rubber tape augmented with 16 fiber optic bend sensors to detect bends and twists to help users create elaborate 3D models.

Since then, new technologies that support flexible displays [118] generally, and even DIY printing of flexible circuits and displays [126, 25] specifically, make it easier than ever for researchers to experiment with bendable devices. The Flex sensors currently available to measure bends[2,3] are also significantly cheaper than their optical counterparts.

---

[1] https://tei.acm.org/

[2] https://www.spectrasymbol.com/product/flex-sensors/

[3] http://www.flexpoint.com/

Flexible displays were not initially available, so early work used rigid displays attached to flexible substrate or a passive surface that is tracked visually so a display can be projected on it. Some early work did not design and evaluate a prototype of a device, but rather elicited users to engage in abstract generation of gestures. In *"How users manipulate deformable displays as input devices"* [102] Lee et al. implemented a user centered approach to determine likely gestures for deformable interfaces. They worked with three materials: plastic, paper, and cloth, of the same size. They asked participants to devise a gesture to one of 11 actions (such as turn on/off, zoom in/out) in random order, then analyzed the participants' agreement on gestures.

Other possible aspects of bending interaction and bendable devices were later explored as well: how does the size of the device affect the interface [103], how does the level of device stiffness affects the interaction [89], and how are the gestures affected if they are preformed with one hand [52]. While researchers have some empirical evidence on these questions, they are far from extensively validated. The potential entrance of commercial devices can change the general population's perception of the gestures and desirable device properties necessitating a new round of investigation. Design, as Gaver says [48], changes the world: *"When the original iPad was designed, for instance, tablet computers were not widely known or available. Now anybody seeking to research or develop tablet computers - or anything at all, for that matter - is designing for a different world, one in which the iPad exists."*

More discussion on bendable devices is presented in the following chapter.

2.2 Input Devices and Techniques

To start a discussion on input techniques, I'll introduce the following definitions from Hinckley and Wigdor's chapter on the subject in the HCI Handbook [66]. "An **Input Device** is a transducer that senses physical properties of people, places, or things. A **Conceptual Model** is a coherent model that users form about the function of a system: what it is, how it works, and how it will respond to their input. An **Interaction Technique** is the fusion of input and output, consisting of all hardware and software elements, that provides a way for the user to accomplish a task, given a particular conceptual model."

These key definitions are at the core of understanding some of the work I present here in regards to bendable devices - as a novel form of input, there is room to explore the sensory technology behind it, the conceptual models formed by people as they encounter it, and the combination of hardware and software, input and output, that form the interaction as a whole.

There are many types of sensor input, such as: motion, range, position, movement and orientation, touch, gaze, speech, gesture, identity (i.e. via a barcode or fingerprint), and brain activity [178]. These forms of input need to be converted into a useful set of instruction that a computational device can interpret and act upon.

Sensors have many properties that a designer must take into consideration. They must identify whether the input is discrete or continuous, the levels of precision the sensor enables and whether the interaction is started explicitly or implicitly. Unfortunately *"many interfaces have been designed that force inherently continuous activities into discrete button interfaces, where in fact it may be more desirable to use a continuous sensor input device."* [137] So, the interpretation of the input signal plays a big role in the interaction technique as a whole.

One way to interpret sensor input is as a set of values that can be translated to another set of values. Perhaps the most formal presentation of this transduction process was offered by Card, Mackinlay, and Robertson [26, 27] in a pair of papers on the design space of input devices. They define the space as a combination of **primitive moves** and **composition operators**, where the primitive moves include linear movement in X, Y, Z and rotary movement around X, Y, Z as either: (1) absolute value change, (2) relative value change, (3) absolute force/torque applied, and (4) relative change in force/torque applied.

They represent an input device as a six-tuple $< M, In, S, R, Out, W >$. Where M is the manipulation operator (i.e. P_x is position change along x, dR_z is delta in rotation around z, dF_y force delta along y, and T_x torque around x), In is the input domain (i.e. ranges like $[0°, 45°], [-Min_x, max_x]$, or sets of values like $[1, 2, 3, 4, 5]$), likewise, Out denotes the output domain; S is the current *state* of the device; R is a resolution function that maps the input set to the output set (for example: a 1-1 mapping for a volume knob would use the identity function $R_z : [0°, 270°] - I \rightarrow [0°, 270°]$, while a mapping of a knob with 3 discrete output values might be $R_z : [0°, 90°] - f \rightarrow < 0°, 45°, 90° >$, where $f(In) = [0°, 22.5°) \rightarrow < 0° >, [22.5°, 67.5°) \rightarrow < 45° >, [67.5°, 90°] \rightarrow < 90° >$); lastly, W is the *Works* and encompass any additional information needed to explain how the input device works (for example, joysticks always return to their 0 position when released by the user, the information of ’return the value of a joystick to 0 if there is no more input’ would be included in the W of the device).

They further define three *composition operators*: merge composition, layout composition, and connect composition. The papers discuss the expressiveness and effectiveness of input devices methodically, however, even the authors agree their framework does not cover the full scope of input devices. Conspicuously, speech recognition (or any gesture recognition) are not covered. Those kind of devices require a sequence and are not simplified to physical movement. However, this is an interesting model to present some simple sensor based systems, such as bendable devices.

A more complex transduction process involves pattern recognition. Wilson [178] includes a discussion on signal processing, feature extraction, and classification and modeling. Using pattern recognition requires a preprocessing phase where numerous signals generated by gestures are analyzed and classified to form the ground truth that helps the system learn. Another way to deal with pattern recognition is programming by demonstration. Hartmann, in his dissertation, [64, 63] worked on creating tools to help author prototypes that use non standard input sensors. One of his tools, the Exemplar authoring environment, allowed authors to demonstrate the signal they hope to get as input. The author would connect the sensors, preform a desirable gesture, then highlight in a panel showing a visualization of the signal which part qualifies as a "valid" input pattern.

It should be noted that pattern recognition based input is akin to the use of commands that pre-date direct manipulation. The whole "command" (or gesture) needs to be entered and evaluated to determine whether it is recognizable or an error. The what-you-see-is-what-you-get simplicity of direct manipulation offers many benefits, such as being reversible and easy to learn [155]. This is yet another consideration for a designer deciding on an input device and an interaction technique. There is some temptation to use "natural" gestures, but it is not clear if natural gestures even exist [72]. On the other hand, on a semantic level, often two or more actions fuse into a single gesture, this is called **chunking** and is useful for lowering the cognitive burden of a task [22].

There are many other aspects of input devices to consider. For example, some devices are direct (touch screens) and other are indirect (mouse). Devices can also offer a mix of direct and indirect input [66].

There is also a wide array of properties that can be measured about an input devices and can be used for metrics. These include **pointing speed and pointing accuracy**, **error rates**, **learning time**, **footprint and**

gain, **user preference**, **comfort**, **cost**, **sampling rate**, **resolution**, **sensor accuracy**, and **linearity** ([66].) Scoltz and Consolvo [148] describe a framework for evaluating ubiquitous computing applications, many of which rely on novel input devices. They offer some conceptual measures, such as, adoption, trust, appeal, and robustness. They also offer some possible metrics, but ultimately, there is no one evaluation solution to fit all systems.

Even after the input device itself is properly designed, interaction with a physical sensor based system can be difficult. Bellotti et al. [11] suggest a framework similar to Norman's seven stages to get over the gulfs of execution and evaluation, but for physical sensor based systems. They raise question like:

- When I address a system, how does it know I am addressing it?
- When I ask a system to do something how do I know it is attending?
- When I issue a command... how does the system know what it relates to?
- How do I know the system understands my command and is correctly executing my intended action?
- How do I recover from mistakes?

It is not always possible to answer these question. Physical and sensor based systems are hard to design.

2.2.1 Events

"[E]vents trigger state transitions which in turn are the building blocks of interaction techniques."[66] Talk about input device state mostly focuses on pointing devices. These mostly follow Buxton's three-state model [24] with the possible states: (a) out-of-range, (b) tracking, and (c) dragging (they are usually marked as states 0, 1, and 2 respectively). Not all pointing devices use all three states; A mouse uses states 1 and 2, while a touch device uses states 0 and 2, see Figure 2.1. Some devices attempt to use all 3 states, such as proximity sensing devices. There is some incompatibility between devices that support different sets of states which makes it hard to design for them. The lack of buttons on touch devices, for example, often leads to awkward "band-aids" like touch-and-hold when applications designed for a mouse are adapted to tablet use.

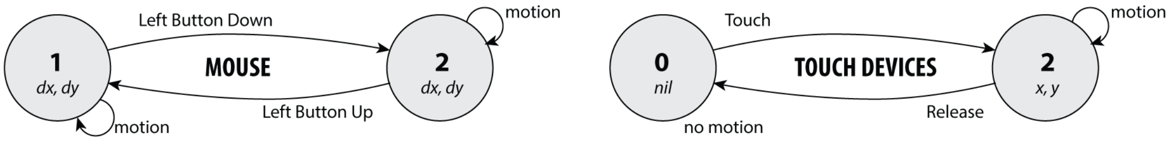


Figure 2.1: Mouse and Touch state transition diagram

In the creation of applications that make use of an input device, application developers use events. For commonly available input devices, operating systems or software frameworks manage an event system to constantly read input, identify all of the events, and make information about events available to applications. Applications, on their part, *subscribe to* or *poll* a subset of the possible events according to the app's interaction scheme.

Events are customarily defined as either low or high level. Low level events correspond directly to states of the input device, for example, a mouse button being pressed down will trigger a **Mouse Down** event, a mouse button being released will trigger a **Mouse Up** event, and a finger touching a touchpad will trigger a **Touch** event. High level events encapsulate more complex gestures or combinations of low level events, for example, we identify a **Double Click** event when mouse-down and mouse-up events occur twice in quick succession, and we identify a **Pinch** event when two fingers touch a touchpad and continuously move toward each other. Having an event model with a wide array of events to choose from and an event system that streamlines working with these events is highly beneficial for application developers. It allows for faster and more flexible code development, and in some cases, mostly for high level events, helps in defining and implementing design guidelines that provide a comparable experience for users across applications.

More discussion on states and events in bendable devices is presented in the following chapter.

2.3 Digital/Physical Hybrid Boardgames

> "The strong attraction of these games contrasts markedly with the anxiety and resistance many users experience toward office automation equipment." -Shneiderman [155]

2.3.1 Digitizing Games

Introducing computers to a game that is not a computer-game seemed inevitable in light of the rise of ubiquitous computing [174]. The idea of **augmented games** appeared, games that maintain characteristics of traditional games while also using the power of computing to add benefits on top of the traditional gameplay. Such benefits included novel game mechanics that cannot be otherwise implemented [110] such as hiding information from players selectively. The augmented games studies by Lungren and Bjork [110] at that time were mostly developed within academia, games such as, 'Can You See Me Now' [44], 'False Prophets' [115], 'MIND-WARPING' [160], Wizard's apprentice [129], and various other examples (see, for example, "Pervasive games: bringing computer entertainment back to the real world" [112]) experimented with groundbreaking technology to augment games - some of them played on a much larger scope than the boardgame category of games that encapsulates card games. Since then, mobile phones and tablets changed the technological landscape (as well as the sociological one) and commercialized some traditional games, blurring the line between traditional and computer games.

However, translating traditional games, specifically boardgames, into pure digital form is not trivial or intuitive. For example, when exploring the **automation** of digital board games [169], which was identified as performing complex or routine in-game activities, acting as an impartial referee, automating game progression, and using digital media to provide a dynamic sensory experience, researchers saw a trade-off between full automation and players' enjoyment. Indeed, we see [180] that players enjoy to some degree the "chores" that are part of traditional boardgames and the social element that they add. In fact, players enjoy the physical components of the games that they could fiddle with [139]. In addition to the difficulty of finding balance between automation and manual player involvement, which Rogerson et al. [138] identify as a tension between sticking to the boardgame metaphor and adding functionality afforded by digitation, there is an added frustration by the companies digitizing these games that resent "sticking to the metaphor" at the expense of taking full advantage of their platform's ability.

And so, from fully digitized boardgames, we look again at augmented boardgames, now using ubiquitous technology like smart-phones and tablets in addition to physical game pieces. As more such games became available commercially, Kosa and Spronck [98] evaluated the attitudes of players toward such games. They saw a combination of negative attitudes from players who resented needing to rely on "an additional thing" to

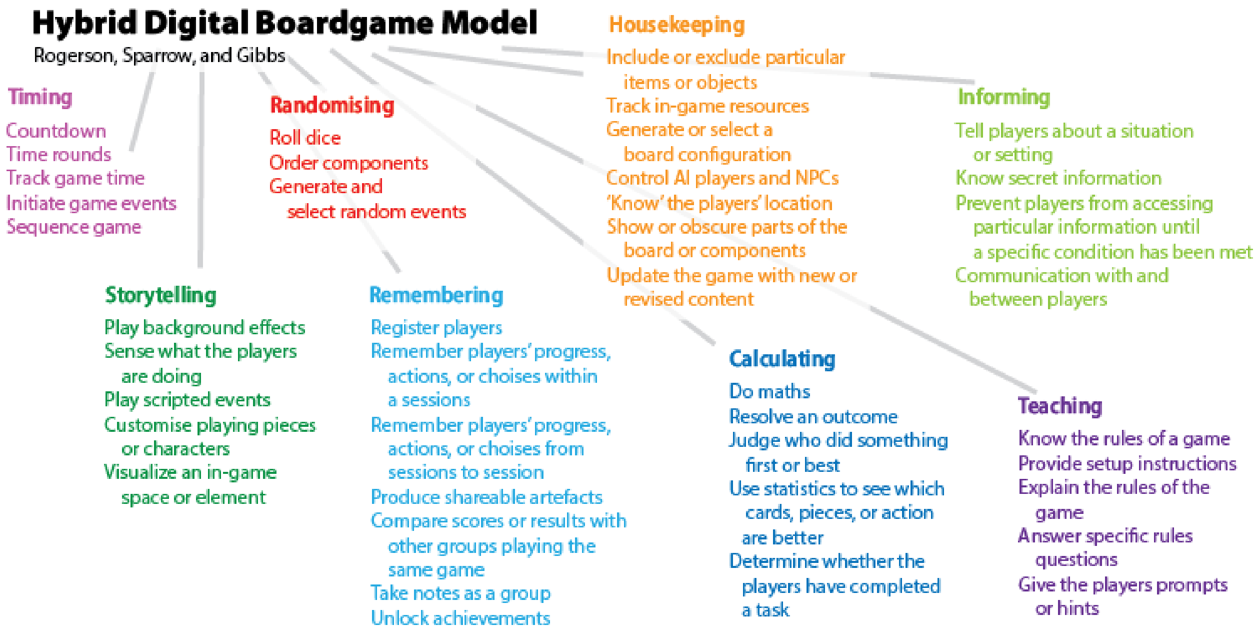


Figure 2.2: Hybrid Digital Boardgame Model

play the game, especially, a digital thing that can run out of power or break down. The more positive attitudes approve of the automation of the game and the potential for novel kinds of game mechanics. Rogerson et al [140] formed a comprehensive model called the "Hybrid Digital Boardgame Model" 2.2 that identifies and classifies all the possible activities digital elements preform in gameplay, these include various automation functions as well as new mechanics. Their definition of **hybrid boardgames** involves *"boardgames in which play is enacted through both physical components and a smartŹ digital element"* requiring that both the digital and physical components are necessary to play the game as opposed to a digital component that is added to the game. I will use augmented and hybrid boardgames interchangeably, as, from the technology standpoint, they do not differ, the difference lies in the intentions of the game designer.

2.3.2 Card Games

While designing for a realization that bridges card-based and fully digital games, it is important to understand what players enjoy, so the source of enjoyment can be maintained. This topic has been better explored in TCGs (Trading Card Games) than regular card games. Adinolf and Turkay [1] found that in both paper and digital realization of TCGs players enjoy similar aspects of the game: card collection, the challenge of deck building and the strong community bonds that form around a specific game.

Sakamoto et al. [144] analyzed TCG players according to three personality types, what they enjoy about playing the card game, and how the switch to virtual card-play affects them. Players with *moving away from people* personalities enjoy the card collecting and manipulation aspect and lose the excitement of opening packs and caring for cards in a virtual environment. Players with *moving toward people* personalities enjoy the social aspect of playing - learning from friends, playing and communicating with each other, helping each other - for them moving to a virtual environment which mostly involves remote playing, loses the main source of their enjoyment. Players with *moving against people* personalities enjoy showing superiority over other players and focus on winning, virtual card games diminishes their ability to read their opponents reaction, and also gives them an easy way out of a losing game - they can close the game.

A switch to interactive cards does not interfere with most of the sources of enjoyment mentioned above. Card collection can remain a valid part of the system if paper cards with special codes are bought and scanned into the system in a similar way to the cards of the TCG *Eye of Judgment* [157]. EoJ was a commercial game (now discontinued) that used a special board, paper cards, and a camera to allow online remote playing.

The scenarios I presented in the Introduction are inspired by past projects and their rationalizations. Early on, Römer and Domnitcheva [141] proposed a set of smart cards with built in RFID tags for playing the game *Whist*. Their setup included a table with RFID readers spaced to capture the table space, and regular playing cards with RFID tag stickers. When a card was placed on the table, it was identified by the system, and updates were sent to a display and personal devices to keep score, provide game tips, and alert players of cheating. Park et al. [127] attempt to increase the level of interest and immersion in the game by enhancing it with card based animations. Their system uses RFID tags on regular paper based cards and a spacial table with both an RFID reader to detect cards and a projector to showcase animations based on the cards in play.

Magerkurth, Engelke, and Grollman [113] confronted another interesting predicament. In their work they describe the Pegasus framework designed to allow combinations of tangible game components with virtual game components. They created physical game components such as an RFID enhanced game board, RFID enhanced cards, sensor equipped wand, and computerized dice cup, and used them in varying possible realizations along with their virtual counterparts in an example game, *Caves & Creatures*. The Pegasus framework supported the gamut of possibilities from using only physical components to only virtual components.

Other projects used cameras or computer vision to virtualize the card game. The Augmented TCG by Sakamoto et al. ([144, 145]) was created following the aforementioned issues players encountered when switching from paper games to digital. They chose to focus on the lack of social interaction in a remote game. Their setup captured the cards of one player with a camera and projected the cards of the remote player on the other part of the table. The novelty they proposed was using an animated character as an avatar for the remote player based on their captured motion to increase immersion and social connection.

Lam et al. [101] harness the strengths of both real world and computer games in *Augmented Reality Table* (ART). Their system employed a TV as a table and an overhead camera to capture paper playing cards used by the players. The system used image processing to detect which cards were in play. A rule engine helped the system decide how to react, whether moves are legal, the effect a move has on the score, as well as using 3D graphics and sound to enhance the experience. This project incorporated the motivation of a rich gaming experience with the practical assistance of gameplay tracking.

A work somewhat similar to my vision for interactive cards by Rooke and Vertegaal [142] described a system that harnesses the idea of ubiquitous displays in a tangible interface of hexagonal tiles. The prototype presented used a projector to simulate the programable displays that each tile would have, and a camera that captured the location of tiles and interaction gestures. The authors demonstrated their prototype with the board game *Settlers of Cattan*. However, the implementation presented requires an elaborate setup, and tile-based games are one of many sub-genre of card play. Of course, it is also possible to create an AR system that uses AR head gear to interpret tangible cards as explained by Billinghurst et al. [13], however, the approach I chose to pursue involves playing cards with embedded displays.

2.4 Research through Design

The literature on Research through Design (RtD), while going back almost two decades, is still heavily involved with defining what RtD is and what is its place within HCI research. Many have tried to classify design contributions in HCI: Horvath [74] identifies 3 methodological approaches of *Research in De-*

sign context, *Design inclusive Research*, and *practice-based Design Research*; Fallman [40] elaborates on *Research-Oriented Design* and *Design-Oriented Research*; Stappers [159] differentiates *Research for Design*, *Research through Design*, and *Research is Design*; Many other names and labels are summarized by Stappers et al. in a comprehensive review of RtD literature [159] repeating the words "research" and "design" in various configuration. So it is a good idea to start by defining both research and design. Sadly, we cannot find consensus here as well. I choose to follow Fallman's definitions [40] where **design** is *"working out the form of something new, consciously creating something which was not previously there"* and **research** is *"to produce new knowledge."* The only agreed upon component of RtD is the production of a prototype, and this could be either fully functional, conceptual, or still at a proposal level [131].

With that in mind, design within HCI research can be parted into
1) research that is about the process of design [4],
2) research that is conducted in order to enable the design of something [5],
3) research that is conducted with the help of something that was designed [6],
and 4) the concept that the designed thing is itself the research (meaning, the designed thing is the new knowledge).

The first part is not quite part of RtD and this review. The second and third seem to trivially fall within HCI research - it is quite common to conduct studies that inform the creation of some prototype, and it is quite common to conduct studies using a newly created prototype. Often in the later cases, the prototype is just a tool for conducting studies and is eventually discarded [108]. The distinction of RtD is that in the RtD methodology, both the studies and the prototypes are equally important as part of the knowledge generated. Hence the concept that the designed thing is itself the research. The questions that arise from that statement are: if the artifact is the knowledge, how is that knowledge conveyed, and how is that knowledge evaluated for intellectual merit?

We have seen other methodologies fight to be acknowledged for their own merits and not through the lens of other methodologies. Famously, Dourish [36] lamented the expectation in the HCI community that ethnographic research will produce "implications for design" (which was also the title of the paper.) He calls

---

[4] this would be Research in Design context mentioned above

[5] this would be Design-Oriented Research and Research for Design above

[6] Research-Oriented Design and Research through Design

to evaluate ethnographic research based on methodology-specific criteria without forcing generalization.

Likewise, design research does not strictly follow the scientific research method. Design often deals with "wicked problems," problems that don't have well defined restrictions, or may even have conflicting purposes [182]. These problems don't have a *correct* solution; If the complexity involved in scientific inquiry is universal, we can say that design complexity is specific [162]; If science opts to be "falsifiable," design is generative, there is no "what is," only "what can be" [48]. Dalsgaard points at similarities between RtD and Experimental Systems in the context of natural science [32], but this is more a testament to the dynamic nature of experimental systems.

It is even controversial whether a RtD project should have a goal, question, or problem that set it in motion. Some [147] declare that design research is guided by a research question. Horvath [74] specifies that practice based design research is *"(i)* ***purposive*** *- based on identification of an issue or problem worthy and capable of investigation*," as well as (ii) inquisitive, (iii) informed, (iv) methodical, and (v) communicable. But others are more flexible. Fallman [40] says that research design is undertaken either to solve a problem, based on theory, or to embody an innovative thought by the researchers, while Zimmerman, Forlizzi, and Evenson [182] state that research in the design discipline often involves creating an artifact that does not solve a problem but rather is meant to stimulate discourse around a topic of inquiry.

Fallman [40] views the incorporation of design into the field of HCI through an historic lens. Disciplines, such as cognitive psychology and sociology were integrated into HCI earlier than design and have a strong effect on the filed's common methodologies. RtD, as an approach that was born inside the interdisciplinary field of HCI, gladly embraces empirical research methods [85, 74]. The empirical process involves phases of explorative research, prototype design, and confirmative research, and is commonly more acceptable in the HCI review community. But not all RtD can follow this process, as Gaver [48] says: *"Seeking conformance to agreed-upon standards and processes may be a route towards disciplinary legitimacy within HCI... however,...such standards might lead to a form of self-policing that would be overly restrictive (of a form of research that I value for its ability to continually and creatively challenge status quo thinking.)"* So how can we evaluate the quality of RtD? The knowledge that results from interaction research is not usually used in design practice, so the knowledge should not be evaluated by how well it is received by practitioners [162].

Zimmerman [182] suggests that evaluation should be done based on **process**, reproducible process with rationale for choices made, **invention**, how novel is the design and how future technological advancements can be applied to improve the artifact, **relevance**, rather than validity, RtD should explain its relevance as a way to come closer to a preferred state of the world , and **extensibility**, what can be built on the outcome of the research. The process can take countless forms including formal experiments, participatory observation, action research, case study, protocol analysis, expert interviews, grounded theory construction, assessment forums, and Gaver's [48] annotated portfolios (which he explain, are the converse to design patterns as they showcase exemplars rather than abstracts.)

There are several reasons I position my work within the Research through Design framing.

Design is Messy The work I present follows a path of possible design of interactive playing cards using bend gestures. This path is not a straight line. The path of RtD is known to be complicated [35] full of blurs, dead ends, parallels, zig zags, loops and more. This is also not the only path, and I make no claims that the artifact I designed is the best. There are many other trails to follow, and perhaps future projects will follow those paths.

Discovery through Design Gaver [48] touts *"making as a route to discovery."* A section in this work is dedicated to expanding the theory regarding events for bendable devices. The question at the base of that particular research emerged while designing the first prototype for the cards. I've reached a point in the software design where I was not sure how to handle the events. I returned to examine the issue in the literature and did not find a satisfactory answer, so I decided to pursue it myself. Without working on a functional prototype, I wouldn't have faced these difficulties. In this sense, the making of the prototype was a catalyst of discovery, which, for me, epitomizes RtD.

Non-Idiomatic Interaction Löwgren [108] describes four ways in which making is significant in interaction design research. Design, he says, frequently involves sketching, and sketching interactions that we are familiar with (idioms) are easier for us as designers, since we can connect in our head what a progression of sketches would look like (for example, transition between wireframes). Bendable gestures are non-idiomatic. Often they are even hard to explain to people unfamiliar with the subject. Interaction design would be impossible without an artifact.

CHAPTER 3
META-ANALYSIS OF BENDABLE DEVICES

3.1 Introduction

Our perception of computers changed drastically over the last few decades. What started as a room full of equipment has now changed into ubiquitous and cheap gadgets found everywhere. At the same time, our perception of computer *interaction* changed as well, from command line input, to direct manipulation with a mouse and pointer, touch gestures on smooth surfaces, voice commands and more. Human Computer Interaction researchers and practitioners are always pushing forward to imagine new forms of interaction. Bendable devices represent yet another incarnation of imaginative interaction design.

In this chapter, I present a deeper view of the current state of research in the bendable devices field. Technical advancement enabled the development of flexible thin displays resembling sheets of papers in their affordance, which motivated the development of paper-like computers, and still inspires new ideas (see Figure 3.1). Some ideas manifested as prototypes - built by research groups to illustrate the feasibility of a device that uses bend gestures and provide some exploratory insight into the use of such devices. These prototypes, how they were implemented, and what they were used for are at the center of this chapter.

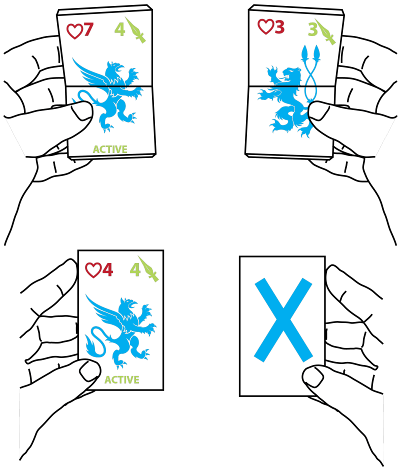


Figure 3.1: Bendable devices - upon bend, the display changes

The meta-analysis in this chapter follows the following form: First, I define **bendable devices** and explain the selection process of the papers included in this analysis. Then, I classify the devices described in those papers into four categories: Digital Paper, Personal Device, Device Periphery, and Foldable - a classification scheme based on the proposed purpose of the devices. Subsequently, for each device, I look at the technology it uses, the implemented or designated applications mentioned for it, its specific approach to bend gesture interpretation, and the research activities described in its related papers. The goal of reviewing these facets, is to observe from a pragmatic point of view what work has been done for the purpose of uncovering what still *needs* to be done to make bendable devices realistic, practical devices for everyday use. At the end of each section I provide some comments on the visible gaps in the field that can be addressed with more research.

3.2 Methodology

Bendable devices, as an overarching class of devices that use bending as a form of input, have not been formally defined, so to clarify the scope of this review I will define bendable devices.

> A bendable device is a device in the intersection of Tangible User Interfaces (TUI) and Organic User Interfaces (OUI) that interprets the bending of the device's surface as input. A bendable device (1) has a display (output) integral to the device, and (2) uses grasp and bend motions to trigger an output change (bend events).

3.2.1 Dataset

The research community for deformable interfaces in general, and bendable devices specifically, is fairly small and the majority of papers mentioned in this review are repeatedly cited in new work. I collected a large body of papers that were cited in known papers or that cited known paper as an initial dataset. I discarded paper that do not describe a bendable device. Some of the papers I include in this review did not have a completed prototype, for example, the prototype may not have an embedded display, but the paper discussed the device as if a display was to be an integral part of it. Several of the papers describe a device, but do not offer user studies.

Overall, I selected 33 papers to be analyzed in this review (see Table 1 for the list of devices and their associated citations, as well as their categories and other properties). Figure 3.2 shows the paper distribution per category per year since 2004.

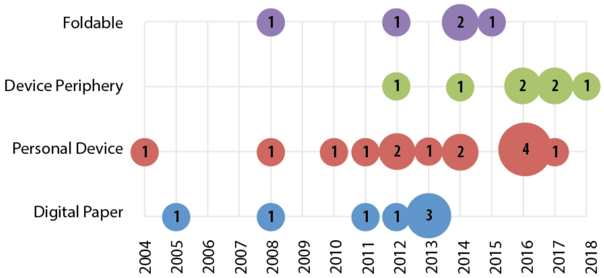


Figure 3.2: Count of the number of papers by category

3.2.2 Exclusions

To focus the scope of this survey I only include devices that are grasped with one or two hands and require a bending motion to trigger some expected action. This excluded several known shape changing interfaces such as Cloth Displays [105] that uses an elastic surface that is not graspable but rather uses pokes and pinches, Bendtroller [156] which uses bends for a would-be game console controller (requiring an external display not coupled to the controller), and Soft and Stretchable sensor array [54] which identifies bends occurring while moving the body using wearable sensors. The MorePhone device [55] was excluded from the dataset despite including bend sensors, because in the user study described in the paper, the participants chose not to interact via bending. Some the selected examples include devices that change the size of available viewing area, however, a device like Xpaaand [87] which changes the size of its display using an ancient paper scroll metaphor to roll/unroll, while graspable, does not involve bending motions.

I also exclude from the collection works that have been highly cited, such as TWEND [65] - which according to the ACM digital library[1] has 28 citations - since it is too incomplete even as a prototype.

In addition, I would like to focus on functioning prototypical devices, therefore this chapter excluded some other prominent works in the field that do not include a device or include a device that is similar to one already selected [102, 103, 91, 89, 5, 52, 31]. These were mentioned in previous background sections.

3.2.3 Classification

In a recent survey paper, *"Analysis and Classification of Shape-Changing Interfaces for Design and Application-based Research"* [164], Strudee and Alexander provide a comprehensive review of shape changing interfaces (SCI), identifying 8 categories of SCIs. However, while bendable interfaces are a **subset** of SCI, the field itself is fairly wide, and the survey favors breadth over depth. Therefore, while Strudee and Alexander cover many of the projects I discuss in this review under their categories of *Bendable*, *Enhanced 2D*, and *Paper and Cloth*, the perspective of their analysis is different from the one expressed here.

In addition, the classification of interaction in Strudee and Alexander's work divides the SCI space to direct, indirect, and remote interaction, then go on to examine what kind of technologies were used for input (deformation, sound, touch etc.) and output (shape, sound, image etc.). In my review, based on our scope - all interaction are of the **direct** kind.

Every paper in my collection was scrutinized with the following questions in mind:

1. What are the characteristics of the device.
2. What technology does the device employ.
3. Does the device include additional forms of input (or are they otherwise implied). If there are, are these input methods used in conjunction with bending or independently.
4. What form of bending does the device accept: discrete, continuous, and/or multiplexed.
5. What applications are suggested or implemented for the device.
6. What type of studies (if any) does the paper describe.

3.3 Categories

After careful analysis of the papers in the collection, I identified four categories that divide the design space of bendable devices: Digital Paper, Personal Device, Device Periphery, and Foldable. The first two categories are evident from the coarser classification scheme reported in the SCI survey paper [164]. Device

---

[1] https://dl.acm.org/citation.cfm?id=1358936

Category	Citation	Year	Name	Display	Gesture	Additional Interaction
Digital Paper	[68]	2005	paperWindows	passive	DCM	Pointing
	[46]	2008	Paper-like input	none	M	Touch
	[100]	2011	PaperPhone	active	D	
	[53]	2012	DisplayStacks	active	D	Relative position, Stylus
	[166]	2013	PaperTab	active	D	Touch, Location on table
	[171]	2013	Bending the Rules	none	D	
	[161]	2013	Flexpad	passive	M	
Personal Device	[150]	2004	Gummi	active	DC	2D tracking
	[165]	2008	Page turning	active	DC	
	[181]	2010	Cobra	passive	CM	Pressure
	[177]	2011	BendFlip	active	D	Pressure
	[90, 91]	2012	KineticDevice	none	C	Twist
	[20]	2013	FlexView	active	C	Touch
	[121]	2014	BendID	none	M	
	[2]	2014	Bendable device	active	C	Touch
	[163, 21]	2016	ReFlex	active	C	Touch, Button, Haptuator
	[56]	2016	WhammyPhone	active	C	Touch
	[116]	2016	Password	passive	D	
	[107]	2017	Bendy	passive	DC	
Device Periphery	[120]	2012	MimicTile	tablet	D	Tablet, SMA
	[135]	2014	FlexSense	tablet	M	Tablet
	[136]	2016	FlexCase	tablet	DCM	Tablet, Pressure
	[6]	2017	Paper for Epaper	tablet	C	Tablet, Electrodes
	[41]	2017	PaperNinja	tablet	DM	Tablet
	[38, 18]	2016/8	BendyPass	tablet	D	Tablet
Foldable	[172]	2008	Bookisheet	passive	DC	Light sensor, Buttons
	[86]	2012	FoldMe	passive	CM	Touch
	[132]	2014	Paddle	passive	CM	Touch
	[57, 58]	2014/5	PaperFold	active	M	Touch, Accelorameters

Table 3.1: Bend Gesture Projects Feature List

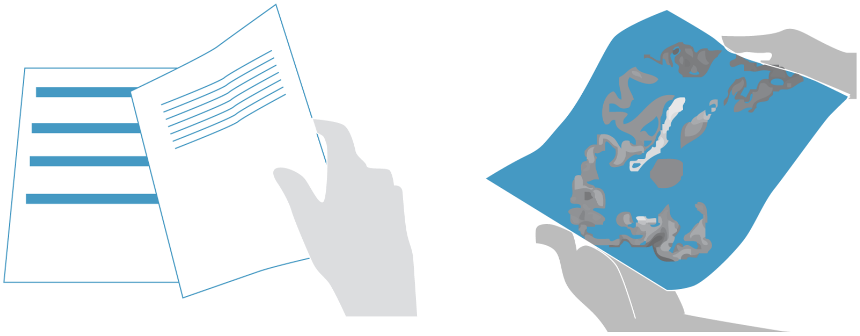


Figure 3.3: Digital Paper

peripheries are bendable interfaces that are attached to an existing smart phone or tablet and therefore use that additional device's display as output. We still consider them under our definition of organic user interfaces since the bendable periphery and the existing device are colocated and appear seamlessly connected. Foldable devices may use rigid displays (but they don't have to) and focus on the interaction of opening and closing panels, fully or partially, as the bendable interaction. This section provides a brief introduction to every device that was categorized based on this scheme. The technologies used for input and output are mentioned and shown in Table 3.1. Illustrations based on exemplar prototypes from each category are presented in Figures 3.3, 3.4, 3.5, and 3.6.

3.3.1 Digital Paper

This category is inspired by the futuristic vision of digital displays becoming so thin and flexible that they haptically resemble paper. Paper, as an object, has many affordances - we write on, erase off, fold, cut, glue, crumple, bend, stack, sort, and staple, to name but a few actions we do with paper, and the thought of combining these affordances with computational capabilities is alluring. Devices in this category often describe their applications domain as document related (reading, sorting, expanding, copying etc.). They are often designed as part of a distributed system of multiple document-devices that are envisioned to scale up to accommodate any number of units. **PaperPhone** [100] is a preliminary prototype intended to showcase multi-device interaction. They introduce the topic of using flexible displays to imitate paper's affordances by listing the positive aspects of paper documents. One such aspect is to *"Have many physical pages, each page pertaining only to a specific and physically delineated task context."* Work in this category has dwindled over the years, however, this category embodies a real-world metaphor that lends itself well to bending and is therefore the one that most appeals to me, serving as the intended category for the interactive playing cards [94].

In terms of technology, this category is split between devices that use passive (non-functional placeholders) and active displays. The active displays used are e-ink displays which have the advantage of being appropriate for document reading, and the disadvantage of slow refresh rates causing slow response time. Designers are aware of this problem, but look toward a possible future where the technology to make fully flexible displays and circuitry with fast refresh rates is accessible and cheap. Likewise, projects that use a passive display assume such a future is forthcoming, and use computer vision systems to track surface positions and a projector to mimic an active display as placeholders for that technology. Most of these devices also support touch interaction in addition to bending, and in fact, some of the interactions require complex combinations of bending while touching.

PaperWindows [68] simulates a windowing environment using real sheets of papers augmented with infra-red reflective markers as passive displays. They suggest multiple types of gestures to handle documents such as collating and rubbing them together. Bending is suggested as a way to flip through a stack of documents. The **Paper-like input** [46] prototype also enhances a card stock with infra-red reflective markers and describes how to recreate the complex deformation of the device. These two projects use a similar tracking system to identify pointing or touching, by instrumenting a user's finger or checking for marker occlusion. **Flexpad** [161] prototype uses a more sophisticated method to detect the deformation of an *unmarked* sheet of paper. Their setup includes an overhead Kinect depth camera which they use to detect both paper and user's hands, computationally remove the hands, and apply a model to simulate the surface. A projector then projects output based on the detected deformation.

Since this category is conceptually grounded in the use of paper documents, the arrangement, layout, and orientations of such "paper" devices was considered an important aspect of interaction in these prototypes. This is somewhat implied with the passive systems that use visual tracking to determine the location of one or more digital papers, but the prototypes using active displays have had to design their devices to also sense location or position based information. **DisplayStacks** [53] uses 3 e-ink displays, each modified to sense bends and stylus interaction. Another layer sports a specially designed pattern of *detection zones* that help detect if the devices are piled (arranged without specific order), stacked, fanned, linearly overlapped (horizontal or vertical) or collocated (side by side). Bending and stylus interactions have different interpretations under the different arrangements. **PaperTab** [166] prototype uses, in addition to bend sensors and a capacitive touch screen, electro-magnetic sensors on multiple PaperTab devices to determine their location on a specially fitted work desk. The idea behind this project is to divide areas on the table into levels of importance, and use the bending to select and load new documents.

The *"Bending the Rules"* paper [171] is a bit of an exception in this category, as the device was mostly built to help classify bend gestures by location, size, etc. However, we include it in this category because its form factor was specifically made to be paper-like.

3.3.2 Personal Device

This category is inspired by the question "what could we do if our personal devices (PDA, phone, tablet) could bend?" While some of the devices in this category enable freeform deformation (with a passive projected display or with no display under the expectation of adding one in the future), most of these devices suggest their affordance by restricting the form factor - offering handles that are suggestive of grasping, directing users to bend in a controlled way.

The older models in this category were developed before researchers had easy access to active flexible display, therefore, they feature rigid displays or even existing e-reader devices mounted on a flexible surface equipped with bend sensors. **Gummi** [150], which is canonically considered to be the first bendable device and is introduced in the visionary paper that established the field, uses a TFT color display attached to a flexible plexiglass base. The base is to be grasped by the user with both hands and bent up or down around the vertical center. There is a 2D tracking sensor on the back of the device to enable position based actions.

The devices in **Page turning** [165] and **BendFlip** [177] also use a rigid device on a flexible substrate with bend sensors attached on both sides used to detect if the user is bending on the left or the right of the device. These devices focus on finding ways to enhance the experience of reading e-books by making the interaction more akin to reading paper books. Users that are regular book readers frequently bend the corner of a page as they flip to the next or previous page, accordingly, these projects mapped bending to page flipping. In addition, readers often bend a book to a deeper extent to provide inertia for faster page

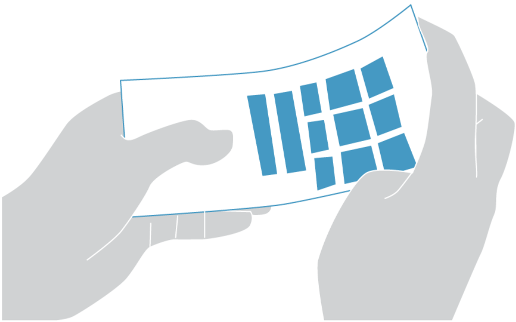


Figure 3.4: Personal Device

skimming or to jump chunks of pages, so *Page turning* device allowed for different degrees of bends.

FlexView [20] is the only device we assigned to this category that uses an e-ink display in a device similar to PaperPhone upgraded to include touch gestures. As time progressed, researchers started having easier access to Flexible OLED displays (FOLED) which have the flexible qualities of e-ink displays, but with vivid colors and better refresh rates. With that kind of display in mind, Kildal's group at Nokia developed the **KineticDevice** [90, 91] prototype which sports two rigid handles and a flexible center that can measure bends and twists using a strain gauge. They used this device (or versions of it) to study various properties of bendable devices and bend gestures. Its successor is showcased in the paper *"What is a device bend gesture really good for?"* - a (**Bendable device** [2]) with a FOLED display in a plastic encasement designed to limit the gestures to center bends. Another research group uses FOLED displays for prototypes like **ReFlex** [163, 21] and **WhammyPhone** [56] which have a less restrictive bendable area, but still use a similar interaction language as the KineticDevice. These devices also include touch interaction.

The last set of devices we sorted into this category **Cobra** [181], **BendID** [121], **Password** device [116], and **Bendy** [107] do not restrict the flexibility of the device's surface, but use bend sensors (or an ITO electrodes array in the BendID case) attached to a flexible foam board to detect bend gestures. Cobra and Bendy were created as gaming platforms and use a passive display while they project a game from the shoulder of the user to the foam board. Cobra uses infrared LEDs to identify the location of the foam board that would be the display, and Bendy has a system tracking a fiducial marker on the back of the surface. Most of these devices settle on having a limited set of acceptable gestures, such as bending specific corners or sides, as opposed to complete freeform deformation (BendID did not go into detail describing

Figure 3.5: Device Periphery

interactions with the device) with the understanding that it is hard to programmatically associate actions with an unlimited set of gestures.

3.3.3 Device Periphery

As personal mobile devices gained popularity a category of bendable devices that would enhance existing devices rather than replace them was formed. This is different than early prototypes of bendable personal devices that used pre-existing devices tacked on flexible substrates due to technical constrains and lack of flexible display availability - the devices in this category are meant to co-exist as periphery attachments to a mobile phone or a tablet. As such, users have access to all the interactions available on the mobile device in addition to the bendable peripheral.

One way to create a periphery device is by adding Around the Device Interactions (ADI) such as the bendable device-case presented in [41] to control the **PaperNinja** game with bends along the 4 corners and the 4 sides. Overall, the authors suggest a dictionary of 11 gestures that control a piece of paper, helping it move across obstacles in a game running on a phone. **MimicTile** [120] is a device that is attached to a smart phone and can both detect bends and provide haptic feedback through dynamic stiffness. They use Shape Memory Alloy (SMA) wires and heat resistant pulleys to control haptic feedback while the user preforms down and up bends as well as a swing. The bending interaction controls the content on the phone - they give as an example zooming a photo in an album - and the stiffness of the material indicates the state of possible interaction, soft material invites bending, and rigid material repels further bending. **Paper for Epaper** enhances the experience of reading on a phone with paper-like affordance. Each side of the device has three sheets of paper with electrodes to detect when the pages are separated, and bend sensors to detect bends. With the combination of these, the authors implement flipping a page in a book or jumping throug a chunk of pages.

FlexSense [135] has a unique approach to a periphery device - using a flexible, *transparent* sheet as a tablet cover. The sheet is modified with specially developed printed piezoelectric sensors to detect deformation, but is still unobtrusive enough to allow viewing the screen through the sheet and interacting with the tablet using a stylus. The FlexSense sensor is translated into a mesh of its surface, so the gestures can be as simple as a corner bend or as complex as a 3D shape. **FlexCase** [136] is a prototype created by the same group that also uses a printed piezoelectric sensor, however it places that sensor on a secondary screen. The sensor can detect both bends and pressure. The authors prompted users to come up with grip and bend combinations and studies those suggestions. In some sense, this device shares some properties with foldable devices as the device-case needs to be folded open to expose the screens, but we consider it still to be primarily peripheral to an existing device. The case supports 4 configurations: book, laptop (the case section is opened down like a laptop's keyboard), back of device (the case is folded around the phone and the user can press and bend it while viewing the primary display), and closed. The secondary display is an e-ink display and therefore suffers from slow refresh rate, so the developers caution that any immediate feedback should be presented on the primary display.

BendyPass [18] is an interface meant to help visually impaired users to enter bend passwords instead of pin codes to unlock their phone. It offers a robust silicone case, with grooves in the material that mark the corners, a vibration component that is activated when one of 10 bend gestures are detected, and a button that can either undo the last entry, or, upon a long press, confirm that the password should be entered. This is an example of a device dedicated to a specific user group and use case.

3.3.4 Foldable

A folding gesture is somewhat different than some of the bending gestures described so far, such as, corner bends or freeform deformation, but it is still a kind of bend and we include here devices that receive the intermediary state of the fold/unfold gesture as input for controlling application state. Foldable devices are inspired by the book metaphor that needs to be opened to view its content and closed to denote it is no longer in use. Foldables also play with screen size, fluidly moving between a small display to a larger one.

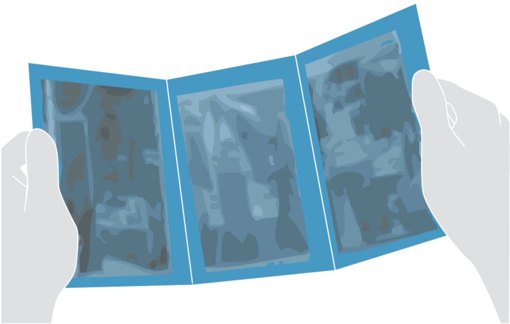


Figure 3.6: Foldable

Bookisheet [172] expresses the design goal of making search through digital information similar to searching through a book. The authors place a high value on the paper-like touch-and-feel of the device that may have qualified it for the digital paper category, however they describe a device with two sheets of plastic connected to look like a book and a light sensor close to the spine of this "book" that identifies when it is open and closed, so that the unfolding of the device seems to be considered a key interaction with the device. Bend sensors along the plastic sheets measure the extent of the bends, which is converted to a jump of one or more pages.

The **FoldMe** [86] prototype assumes future availability of displays that are so thin (in addition to flexible) that a device's panel may have a double-sided display. They suggest fold-to-back and fold-to-front as potential gestures, as well as keeping a finger wedged between the panels like a finger in a book. They suggest three formats for a potential device: using a book fold (two equally sized panels), a partial fold (one of the panels is narrower than the other), or a dual-fold device where there are effectively 3 panels. The prototype also supported touch. Both folding and touch are detected by Optitrack motion capture cameras tracking IR retro-reflective markers.

Paddle [132] system uses a similar tracking system to FoldMe's to track touch interactions and the changes in Paddle's highly deformable shape. The device is built upon Rubik's magic puzzle tiles[2], which are connected in such a way that allows transforming between a small size panel to a larger size panel, to a loop of connected tiles, to a hinged leaflet-like configuration and other forms. The scenario of usage they propose involves a user getting a phone call and then switching between the Paddle topologies to see items

---

[2] https://www.rubiks.com/rubik-s-magic.html

on a larger screen, such as a map, scroll through a loop to see multi-rowed data of weather information, or leaf through items in a calendar.

While Bookisheet, FoldMe and Paddle used a projector to show content on a passive display, **PaperFold** [58] uses active e-ink displays on three detachable tiles. The tiles connect with magnetic hinges. Sensors to detect magnetic field and orientation in space are embedded in the substrate of each tile and can distinguish between a variety of shape configurations the authors elicited from participants.

3.3.5 Summary

Four categories of devices are described in this section as well as the properties that distinguish them. The Digital Paper devices frequently used multiple paper-like sheets, each representing a document. The Personal Devices had a form factor similar to a phone or a tablet, often used in a landscape mode, supporting applications like e-books, maps, and games. Device Periphery devices are meant to accompany existing smart devices. And Foldable devices are hinged devices that can open and close using folding gestures around those hinges.

We can see from the timeline in Figure 3.2, that the categories trend at different time intervals. This is perhaps not significant due to the overall short time-span of bendable devices research. However, we can find some correlation between the visible trends and the rate of mobile devices and smart phones adoption rate by adults living in the US published by Pew Research Center[3]. When work on bendable devices started in 2004/5, more than half of adults in the US had a cellphone, which indicated that the populace was open to wide use of personal devices. New devices were introduced at a fast rate and it is plausible that researchers wanted to explore new interaction modalities. By 2007, smart phones were introduced and work on clever interaction for mobile devices propelled more interest in bendable devices. By 2012, nearly half of adults in the US owned a smart phone, clearly, smart devices were here to stay, they offered vast capabilities, reliable touch and accelerometer sensing, and numerous applications. Parallel to the continued research into turning these smart devices into bendable devices, the category of device periphery formed, possibly with the assumption that smart devices have such a large following, it is not practical to expect the populace to abandon what they know for a new kind of device, but fairly easy to offer them an accessory device that will

---

[3] https://www.pewinternet.org/fact-sheet/mobile/

enhance their current one. The versatility of tablets, which started gaining their place in the workspace, may have nudged the community away from work on digital paper. Foldable devices were floating around as a way to potentially harness existing devices, but find creative ways to modify their screen size on demand, so it could shift from personal phone sized to larger tablet sized display. Over time, rumors of foldable phones started spreading, and it appeared to be an inevitable commercial product, prompting researchers in academia to suspend experimental foldable devices.

3.4 Applications

This section covers the applications mentioned in the selected papers. Some projects have implemented these applications to demonstrate the capabilities of their device or to use them in user studies. Other works provide a detailed or cursory description of what kind of applications could be made for their device and how users would interact with those applications. Here, we collected all of the applications we found in the papers, sorted them by similarity, and divided them to application groups: *E-reader, Navigation, Selection, Zooming, Context & Detail, Layering, Text entry, Value control, 3D control, and Games*. Table 3.2 lists devices based on the applications they associate with (and arranged by device category.)

3.4.1 E-Reader

Bendable interfaces, especially in the form of e-ink displays, are so reminiscent of the paper/book medium that the E-reader application is easily the most common one to be mentioned in papers. This application involves browsing through pages of text, paging forward or backward either one or multiple pages at a time.

3.4.2 Navigation and Selection

There are common interactions dedicated for navigation and selection in WIMP environments, such as, using arrow keys, scrolling, clicking or double click to make a selection. The touch interaction paradigm offer parallels with swipe and tap gestures. In this group of applications, developers tried to apply bending for these tasks.

Since many of the devices in this review also included touch interaction, the rationale of preferring bends over touch for these actions is not always well argued.

3.4.3 Zooming and Value Control

From the inception of bendable devices with Gummi, developers were attracted to the continuous aspect of the bending gesture and its built in bidirectional action - intuitively, the opposite of bending up is bending down. As a result, using bends for the complementary actions of zooming in and zooming out is often featured in connection with bendables, usually associated with a map application. MimicTile used zooming in and out in a photo browsing application and used the shape changing properties of the device to harden it as indication that the zoom level is maxed out.

Likewise, having a slider-like continuous value that is controlled and changed with a bend was used in various applications. In Bendy, for example, a prolonged bend could effect a continuous change in position or size of a character on the screen. In FoldMe, the degree of bend would control a value like the brightness of a picture in a photo viewing application.

One of the questions that was brought up by multiple studies, was the comparison of changing a continuous value by positioning control, or using a rate control. More on this topic below.

3.4.4 Context & Detail

Digital paper type devices and Foldable devices are particularly suited to implement context plus detail type applications, since they have a visual division into multiple devices or multiple flaps, which helps with the contextual division. PaperTab describes an interaction to load a new document by using a paperTab device to bend-touch another paperTab that would hold the menu of available documents, the bent device would then load the selected document. DisplayStacks would show contextual menus based on the content of the device they overlap. The FlexCase device offers a secondary screen to a personal device with the intention of it providing contextual information to save space on the primary screen.

3.4.5 Text Entry

Text entry application for bend gestures was presented in the Gummi paper. The authors suggested two methods of text input, one with layers and one with nested grids, that have some complex interaction with selecting a character with 2d tracking, transitioning down and transitioning up to select, or a combination of these. This application was not a success. Text entry was abandoned as a possible application for bend

Category	E-reader	Navigation	Selection	Zooming	Context&Detail	Layering	Text entry	Value control	3D control	Games
Digital Paper	PaperWindows Paper-like paperPhone DisplayStacks	PaperWindows Paper-like paperPhone	PaperWindows Paper-like paperPhone	PaperWindows Paper-like paperPhone	PaperWindows DisplayStacks PaperTab	DisplayStacks			FlexPad	FlexPad
Personal Device	Page Turning BendFlip FlexView Bendable Reflex	KineticDevice Reflex	Gummi KineticDevice	Gummi KineticDevice FlexView Bendable			Gummi Password	KineticDevice Bendy	BendID	Cobra BendID Bendy
Device Periphery	FlexCase Epaper	FlexCase Bending Blindly	FlexCase Bending Blindly	FlexCase MimicTile	FlexCase	FlexSense	BendyPass		FlexSense	FlexSense FlexCase PaperNinja
Foldable	Bookisheet Paddle	FoldMe Paddle	FoldMe PaperFold	FoldMe Paddle PaperFold	FoldMe Paddle PaperFold	FoldMe PaperFold		FoldMe	PaperFold	

Table 3.2: Common application types

gestures until the idea of using them to input password - not unlike the gesture lock screens common these days - came to be. This idea was especially touted as an assistive interaction for vision impaired users.

3.4.6 Layering

An interesting and underutilized application of bendable devices is to control layers of information on top of some content. FoldMe describes a map application using a dual-fold device, where folding one flap enhanced the map with textual information and the other flap with a set of points of interests. In addition to supporting switching layers in Photoshop or map applications, FlexSense beautifully describes its inspiration from the animators' desks where sheets of paper are placed on a light table to help copy the lower layer with the required modification. The FlexSense sheet can be partially lifted to expose some modified information (for example, the solution of a cross-word puzzle) in "another" layer, though it is all maintained on a single tablet screen.

3.4.7 3D Control

Devices that can detect fully fluid shapes like FlexPad, BendID, FlexSense and PaperFold afford making a freeform shape that can be used for advanced 3D control. BendID authors suggest that it can be used for a 3D modeling application. Flexpad created an application that helps users create a curved slice through 3D volumetric data, an application which is particularly useful for analyzing important medical phenomena. A button activated by foot can lock the view of a slice after a researcher found an interesting cross section that they want to further explore.

3.4.8 Games

Game applications are amenable to new interaction techniques, as is evident by the invention of many novel game controllers - from gloves, guitars, and steering wheels to motion based controllers, users are willing to try new things if they fit the games they want to play. Some of the bendable devices suggest using form to imitate a complex 3D shape such as flapping wings, or to use their flexibility to spatially map game action to the device's haptic feedback, for example in a racing game where the direction and degree of the bend may map to a direction of movement and velocity. Some works pair bend combinations to existing game actions like moving, jumping, and shooting. The gestures and gesture interpretations in this category are mostly covered by the other applications with this category offering a contextual classification.

3.4.9 Summary

It is not surprising that E-Reader applications are the ones most frequently implemented for bendable devices - it comes very naturally from the imagery of the reader holding a book and bending the corner of the page they intend to flip over. Using bends for zooming maps or pictures is both common and intuitive as it can be imagined that curving a surface toward one's eyes will make the content on the surface larger, and curving it away will make the content on the surface smaller.

If there are benefits of adapting to bend gestures applications that are currently operated with touch gestures, like navigation and selection, it is not yet clear. The application of text entry, which is regularly used on devices, was tried with bend gestures and was promptly rejected at early stages, though some recent work is testing its applicability as assistive technology.

Playing to the strengths of bendable devices with advanced applications of context and detail, layering, and 3D control appear to be promising, but requires further study. It is possible that future designers will discover novel and unthought of applications that are better suited for bend gestures than other interaction schemes, and it is my hope that the PEPA project can be instrumental in the creation of such designs.

3.5 Gesture

Detecting a (usually) sensor based input signal such as bend is not as simple as detecting a button or key press. For a button, the interaction is binary - it is either on or off. When the button is pressed, a circuit closes, and it is up to the driving software to notify applications when a press is first detected, that the button is down for as long as it is, and when the press has been released. These are common types of *events* generated in the realm of button interactions. Bend sensors, as many other sensors, have ranges of values. Moreover, their sensing quality may deteriorate over time, requiring calibration to new values, and there is no guarantee that a user will hold bendable devices in their proper neutral state when they are not interacting with them. This complicates the interpretation of bend gestures. In addition, freeform deformation leads to even more interpretations - what action should occur when a certain shape is formed? How do you define these things?

This section goes over the various methods described in the selected papers to interpret bend gestures. Sadly, not many papers detail their event model - how they determine when and what gesture occurred. I identified three main approaches to bend events: the **discrete** approach treats bends as if they represented binary actions like a button, the **continuous** approach looks at the range of possible bend values on a continuum, and the **multiplex** approach can have complex combination of gestures like touching while bending, or deforming a surface into a random shape. The gesture column in Table 1 indicates my interpretation of gesture classification for each reviewed device, though some of these are speculations based on device description rather than based on explicit discussion in the corresponding paper.

This discussion of gesture interpretation and event identification is particularly pertinent to my work on PEPA and understanding the mental model users form for bends. Since I find this topic crucial for the practical design and development of bendable devices [93] it is unfortunate how little it is studied; There is no standard or consensus, yet most papers don't mention their event model.

3.5.1 Discrete

Gummi, as the pioneering paper in the field, offers a model for bending states and events. The authors note that the natural oppositional nature of bending up and bending down exemplify how bendable devices work well for mapping an action and its opposite. The top sketch in Figure 3.7 shows a rough outline of the model they suggest toward the "up" side (the "down" side is similar.) While there is no bend - the state of the device is **neutral**, while it is bent up it generates a continuous event of **transition up**, and when it is bent to its maximum, it triggers a discrete event called **target up**. **Transition down** and **target down** are similar events in the down direction. After a target up/target down event has triggered, the device must fully return to neutral state before another target up/target down event can trigger. A quick succession of target up or target down events are called Double up and Double Down and are likened to a double click on a mouse.

The Gummi model, in essence, defined continuous events that culminate after a threshold is reached in a discrete event. This discrete event cannot be recreated until the device fully returns to neutral mode. Using a threshold makes sense as a mechanism to convert sensor output that spans a continuous range of values into discrete gestures. However, this approach is not as straightforward as it seems, which results in different variations throughout the literature.

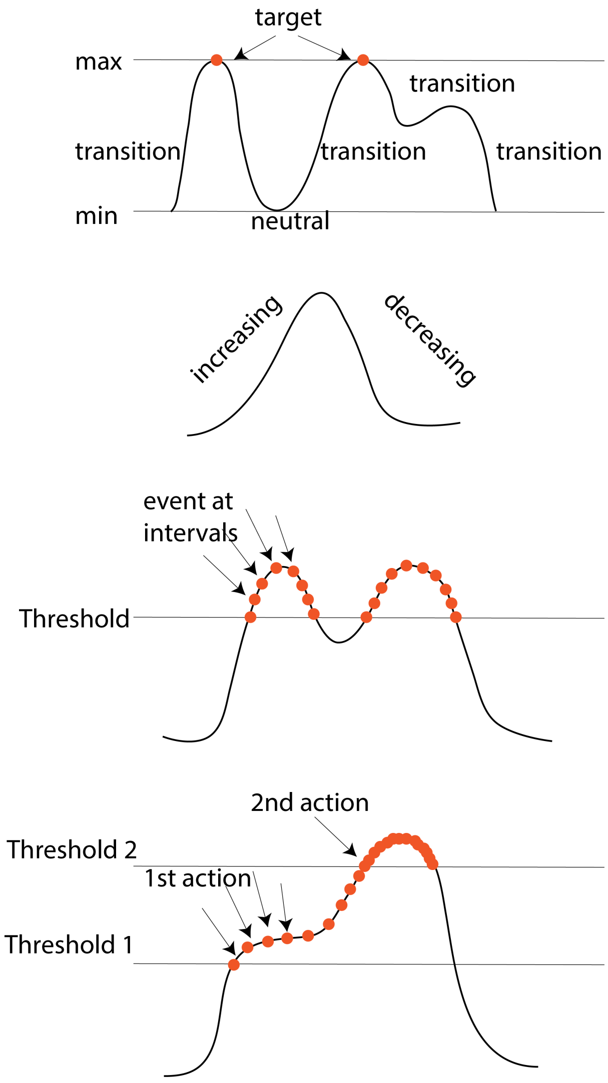


Figure 3.7: Approaches to interpreting simple bends

The creators of Bendflip had undergone several design iterations figuring out how to detect the bend events. Initially, they wanted a simple 30 degree bend threshold, but they noticed that the signal patterns are not that simple and that users don't always return the device to the resting state. As a result, they decided to consider the threshold value as a relative displacement from a recent "resting" reading. They describe their algorithm to resolve bend gestures *"Final algorithm creates upwards (or downwards) flick events when the edge of the device is bent at least slightly upwards (or downwards) after a peak deformation occurring within a window of one* *second."*

Another variation combining continuous and threshold approaches involves triggering events at constant intervals while the bend is past a threshold point (lower two sketches in Figure 3.7). In Bendy, a bend event is triggered when bend sensors' values indicated a bend above a threshold of 30 degrees, and the state of the bend is continuously polled, so *within* the threshold based binary action, the event is continuous. In the paper describing the Intuitive Page-Turning device, the authors decided to use two threshold values for small bends and large bends subjugated to timed intervals tests. Depending on the input values and state in time, the device triggers different events.

Even in a fairly simplistic situation where a gesture is to be interpreted as a discrete event, some devices were trained using machine learning techniques to learn to identify gestures. However, this is more commonly seen in multiplex gestures and will be expanded on in that section.

3.5.2 Continuous

In *"What is a device bend gesture really good for?"* [2] the authors state that the advantage of bend interfaces is in aspects of tangibility, directionality and metaphor. From that perspective, bend gestures should be used for continuous interaction based on the sensor's output value. A pilot study they conducted between touch and bend gestures for actions of horizontal slider, list browsing, zooming, map navigation, and web browsing showed that participants preferred bends for zooming and lightly for list browsing, but for the rest of the actions, touch was preferred. Similar results show that using bends to replace common touch (or click) interactions may not be promising. Prototypes like FoldMe used a continuous folding gesture as spin control, but the authors incorporated touch interaction with the folding (though they admit there may be false touch events).

While presenting the KineticDevice prototype, Kildal et al. shared a road map to the future research activities in the bendable devices field: *"We need to understand how various factors influence the accuracy and controllability in performing deformation gestures as well as the user experience."* One factor that they suggest investigating, is whether to map deformation in a continuous gesture to absolute position (**positional mapping**) or velocity displacement (**rate-based mapping**) of an element on the screen.

The position control vs. rate control question, later explored on the KineticDevice, was also studied by the ReFlex group. Specifically, the ReFlex device featured an haptuator which generated pulses, and the researchers combined absolute and rate based feedback pulses with the absolute and rate based positioning of a target on the screen. Generally, the rate control seems harder for users to manipulate.

A last point of concern with continuous events is how to determine when a desirable value is reached. For example, if a user tries to control a value of some property with a bend gesture mapped to a slider, how does the user set the value of the slider mid-bend. The solution in the ReFlex prototype was a button the user could hold behind the device and click when a value was reached, and in FoldMe the user would touch a menu item to select the current value mid-fold. The kinetic device asked participants to dwell on a value for a second to make a value selection, however, dwell time can be a tricky option as it can be difficult to keep a bend exactly as-is for a prolonged duration of time.

3.5.3 Multiplex

Despite several caveats mentioned in the sections on discrete and continuous gestures, it is fairly easy to identify the signal of a single bend sensor. Even devices that use, for example, bend gestures on all four corners, will usually have a bend sensor dedicated to each corner to separate the complexity of the problem. However, some bending gestures can be more complex, for example, creating a wave shape with a flexible sheet. Other gestures may involve several interaction components, for example, PaperWindow has multiplexed interaction such as staple - bring two papers one toward each other like a clap. Not all multiplex gesture devices explain their gesture interpretation, especially when combining bend gestures with touch. The ones that do provide details, often use machine learning or clustering algorithms.

The PaperPhone device uses kNN algorithm with k=1 to learn gestures entered by participants, and MimicTile uses an artificial neural network (ANN) specifically trained on gesture input. The FlexCase work focused on defining grip and bend combination gestures, they asked participants to come up with promising combinations and devised a learning algorithm that uses a sliding window over the last sensor reading to identify the preceding grip and the bend. The FlexSense paper contributes two algorithms to reconstruct the surface geometry of the flexible substrate based on a new sensor layout of the piezoelectric sensors. One algorithm uses linear interpolation and the other uses machine learning. To train their system, they installed a camera system to capture and identify the 3D deformation using markers, and used that dataset as ground-truth.

BendID consists of four layers: a silicon layer, an ITO (Induim Tin Oxide) electrodes array, conductive foam, and conductive fabric. The electrodes are arranged in a 3 by 3 matrix on the top and bottom of the foam device. When a bend is sensed, an Arduino controller reads the sensors in two order sequences to form a pattern. The system used an SVM model to detect a set of 16 specific patterns they have trained their device to recognize. At the same time, the input values are sent to calculate the magnitude of the bending - the authors have measured samples with different levels of bending, used a polynomial regression model to find an equation to represent the magnitude, which they say is 91% accurate.

Another complex system is detailed in the FlexPad paper. The authors first explain how they use the Kinect infra-red depth camera to remove the hands and fingers of users from the captured image in a process called *optical surface material analysis* - the Kinect camera shows materials with distinct reflectivity and translucency properties such as skin and foam differently. Then the authors explain the model they used to simulate the surface and detect deformation. They divide the plane to 25 by 25 vertices and define 8 basic deformation, a z parameter, and 6 DoF for 3D transformation to define 15 dimensional vector that define their model, the paper then details how they perform their calculation in real-time. This paper is frequently cited for its algorithmic contribution.

3.5.4 Summary

Hopefully, I have made it evident in this section that the sensing or analysis technology used for bend gesture detection is key to bendable devices' capabilities. New sensing developments and new algorithms can fuel future devices. It is perhaps the case that even the simpler devices will need to "learn" their users' gestures to accommodate different grip and bend strengths. In any case, the lack of consistency exposed in this review shows that there is room to create a unified model of possible bend events [93] that will frame the communication around bend gestures to a known vocabulary that both users and application developers are familiar with, like the vocabulary that formed around touch interfaces.

3.6 Research Activity

This section will not enumerate all of the research activity in the field, but rather, give characteristic examples of the kind of research seen. I broadly divided the research activities into *plausibility* of bend gestures, the *difficulty* of using them for specific goals by the users, the *accuracy* of the proposed system in identifying a gesture, the *usability* of bend gestures compared to other interfaces, and the *learnability* of using deformable interfaces.

3.6.1 Plausibility

As a young form of interaction, the field of bendable devices is still fumbling about trying to determine what *kind of gestures* can be done with bendable devices and what kind of applications or actions would *be suitable* for these gestures. Studies in this direction can be found in the papers for: PaperWindows, PaperPhone, Bending the rules, FlexCase, Bending blindly, Bendy, what are bend gestures good for, Bookisheet, and PaperFold.

For example, the design of PaperFold was inspired by capabilities of smart phones but with paper-like affordances to fold and tear apart. For that reason, they created several tiles that can be connected, disconnected, and folded around a magnetic hinge. In a participatory study, they asked 15 participants to interact with the device when it has two tiles, then with three tiles, and finally with four tiles, and come up with as many configurations of displays as they can think of, and what each configuration would present. They created a dictionary with the most popular shape configurations with two and three tiles and remark that the four tiles device was uniformly disliked.

In the Bendy paper, the authors wanted to assess the appropriateness of a set of 20 bend gestures to game related actions divided into: navigation (move up/down/left/right), action (shoot, jump, rotate), and deformation (stretch, squeeze, and change size). In one study that the group performed, 24 participants were shown an effect, such as a character moving or stretching, and were prompted to choose a gesture that would have caused that effect. They measured the agreement between participants. They concluded that deformation showed the highest agreement ranks while the action category had the lowest agreement, meaning it was hard to naturally map a bend gesture to common game activities like shooting and jumping.

3.6.2 Difficulty

One acceptable way to measure that a device has a reasonable level of difficulty is to validate its compliance with Fitts' law [43, 111]. The Fitts' law model is used to measure the throughput of a device (in bits per second) which combines into one measure the speed and accuracy of users' performance on the device. The law provides an equation that ties the movement time required to reach a target with width W at a distance A with an index of difficulty ID.

$$MT = a + b\log_2(\frac{A}{W} + 1)$$

Several variations on target acquisition tasks were presented in the papers on Bookisheet, BendFlip, and KineticSevice, but the most robust evaluation is offered in the ReFlex related paper by Burstyn et al. [21]. In this paper, authors presented an experiment to investigate the canonical one-dimensional targeting using bend input. Participants completed an ISO 9421-9 Fitts' law experiment using two input mappings: position control and rate control, and three levels of display stiffness: soft, medium, and hard.

The devices were similar in shape to the ReFlex device, using substrates of three stiffness levels (an independant variable in the experiment), with rigid handles for grasping on both sides of the screen. They were calibrated so a similar amount of force was needed to get the same bend effect in all three devices. For the experiment, they created 12 ID combinations (where ID is index of difficulty, conditional to the target width and distance between targets) and asked participants to perform target acquisition with 25 repetitions in each block.

Overall, they had 12 participants with each one performing 1800 recorded trials. They built regression models per each participant to check correspondence with the Fitts' law and found very high correspondence. They also note that position control had better throughput than rate control in all conditions, and was selected by participants to be more efficient and easy to use. They did not see a significant effect across levels of stiffness when the amount of force is similar across devices.

3.6.3 Accuracy

Since some of the selected papers deal with new sensing technologies or novel gesture identification algorithms, they include studies that show how the researchers evaluated the accuracy with which the depicted device can recognize gestures. Accuracy related studies were discussed in PaperPhone, DisplayStacks, Flexpad, MimicTile, FlexSense, FlexCase, and BendID.

The FlexSense paper develops two algorithms to learn the surface deformation and identify it, one using linear interpolation and a second one using regularized least squares (RLS). They created a dataset of ground-truth data using marker-based vision systems, they used both a one camera rig and a stereo camera rig - the single camera rig is known to have reconstruction errors which they could later compare to their own algorithms. The total dataset they created included 40,000 frames of deformed surface configurations. They randomly split the dataset into training and validation so that 30,000 frames were used for training and the remaining frames were used for testing. With the results, they calculated the errors and were satisfied that their algorithms were more accurate then a single camera marker-based system.

3.6.4 Usability

Usability is covered by a wide array of studies, but in this section, I refer to studies where users were asked to perform a realistic task while the researchers observed their activity. There is surprisingly little in this group of studies, perhaps due to the immaturity of the field.

In Bendy, 12 participants were asked to play six arcade game: pong, bricks, pacman, tetris, space invaders, and fat cats (inspired by angry birds), on two sized of display and provide feedback about their user experience.

Paddle authors compared their device in peeking mode (flipping a page down or up,) in a searching task, and in the scrolling mode (turning a band of connected 8 tiles around) in a word encoding task, and in the leafing mode (flipping a tile while the device is in a leaflet format) in a fact verification task. These were compared to similar actions on a touch based paddle device. They did not want to use a standard touch device to remove possible confounding factors caused by the paddle display and detection system. However, they got mixed results that led to a need for further studies.

The Password project compared password entry with a bendable device to pin entry on a conventional phone. The 25 participants were asked to re-enter the passwords they created in a second session a week after the first. Generally, participants needed more time to create and enter the bend gesture, though that may be in part due to using previously known pin codes, the feedback was mixed with participants not strongly preferring or disliking bend passwords.

3.6.5 Learnability

When new interfaces pop-up, it is interesting to learn how quickly and effectively users can learn to use them. If an interface is too hard to learn it is not likely to be favored by most users. There is not a lot of work yet on this topic, but the Password project engaged in learnability, and the device called PaperNinja comes from the paper "Effects of Bend Gesture Training on Learnability and Memorability in a Mobile Game" [41] which focuses on learnability.

Held with both hands in landscape, the available gestures included bends in four corners and four sides. A collection of 11 gestures is programmed to help a piece of paper (PaperNinja) move across obstacles in the game. For example, bending the top-left and bottom-right corners would crumple the paper, bending the top-right corner would shoot, and bending the whole top side would fold the paper in half. In the study they looked at three groups divided by no training, gesture training, where participants are taught the gestures but not how they relate to the game, and mapping training, where participants are taught the gestures and how they relate to the game. They had 30 participants overall (divided into the three groups) that attended two sessions each to evaluate the memorability of the interaction. The results showed that no training and mapping training did well in learnability and memorability, while gesture training did not do as well. This may be another indication of the difficulty mapping game actions to bend gestures.

3.6.6 Summary

An overview of the research activity in the bendable device field shows that there is much to be desired. There is especially need for more usability studies that test devices in probable usage scenarios. There is also room to step back and ask: do we need bend gestures to replace touch gestures? Some applications try to recreate touch interactions in bendable devices, but there is no strong evidence to show that this is desirable. It may be plausible, easy, accurate, and easy to learn, but desirability is a key component for a practical interaction paradigm. I suggest to focus research activity on potential strengths of bendable devices. I intend to do so by shifting the context of use to an environment that does not copy existing touch systems, such as PEPA.

There are some other technical points to consider from an HCI perspective, such as, how does the system recover when a gesture is mistakenly identified, how does the user get feedback about what was the identified gesture, what kind of standard gestures can we define that users can expect would work the same across devices. Hopefully, bendable devices will become more of a reality and the research community will devote itself to a wider variety of studies.

3.7 Conclusion

This chapter on bendable devices is intended to provide a view of the current state of affairs in the bendable device field, as well as draw observations on the less explored points.

CHAPTER 4
OBSERVING CARD PLAYERS

In this chapter I present a user study that involved observing card players during natural game-play to learn how players interact with conventional paper cards. This study joins the meta-analysis presented in the previous chapter to form the *suitability* aspect of my research agenda, and, more specifically, RQ2 *How are cards bent in a real-world game scenario?* According to the **sensing-based interaction framework** described by Benford et al. in [12], a sensor based system is defined by the *expected*, *sensed*, and *desired* gestures, where the desired gestures are those we want the users to perform, the sensed gestures are those that the system can identify, and the expected gestures are those that the users are likely to preform whether they are desired/sensed or not. During my observations and analysis, I noticed that the constant contact players have with the faces of the cards may present a deterrent to use touch interactions, while the natural occasions of card bending coincide with meaningful game actions, making bend interaction a plausible contender for interacting with digital playing cards.

4.1 User Study

The study's objective was to gain a better understanding of how players touch paper cards naturally while playing. During this study I recruited 2 pairs of players and observed them during a play session. I video-recorded their hand gestures and analyzed the interaction between the players and the cards. By observing players' natural gestures, I was hoping to understand what the *expected* gestures for digital cards might be, so that I could make sure that the *sensed* and *desired* gestures overlapped with the expected.

4.1.1 Participants

I recruited 4 participants. They were divided to pairs, and each pair had one session where they played several card games of their choosing. The participants were recruited among the graduate students in our department. The participants knew their partners and had played with them in the past. All participants indicated that they play often (every month) and have very good card game dexterity skills (such as shuffling).

4.1.2 Procedure

I met with each of the two pairs in a lab setting and explained the study to them. I informed them that the sessions would be recorded and started recording with their permission. They were asked to fill a survey about their skill with card games and list card games they play. Next, they started playing rounds of games of their choosing. They were asked to play at least one game with a regular (4 suits) card deck as well as a game using a non-conventional deck such as a TCG (Trading Card Game). The games that were played are: go-fish, crazy 8's, Star Realms, and M:TG (Magic: the Gathering). The players were given a standard 52-card deck to play with, but they brought their own specialty game decks with them - players usually need to know their cards and have the ability to edit their decks in such games. At the end of the sessions, the participants were given a $5 gift card in appreciation of their time.

4.1.3 Data Collected

I used two cameras. One camera was positioned over the play table facing down to record the table surface and the players hands from the top, the other had a wider side-view of the table and players' torso.

I collected an hour and 27 minutes of footage, and used the ANVIL tool [92] to process the videos. I made a preliminary coding of the material in search of key categories, and then tagged the videos in accordance with these categories. I was particularly interested with the way the players' hands interacted with the cards: how they were holding them as a single or collection of cards, and how they transition from one form (such as, fanning cards, placing on the table) to another.

4.2 Results

I was interested to see how players manipulate cards: players either hold multiple cards "fanned" to show a glimpse of the card (fan), or they hold multiple cards gathered together as a stack, or they hold a single card. Finger positioning can vary on single or stack hold - a grasp might place the thumb of the front or back face of the stack/card and the other fingers pincer the other face, or the grasp could orient on a parallel pair of sides (top-bottom, right-left) - a fan grasp places a thumb at the front-bottom corner of the cards and the rest of the fingers grasping the back-bottom face of the cards.

Anvil supports multiple channels along the timeline of the video, each channel defined as either discrete or interval, and each channel can have attribute/value pairs associated with it that can be set at each event. After the initial coding and familiarity with the material I chose to code it by these properties:

- Per player (dubbed player A and player B) I created 3 channels
  - Right hand
  - Left hand
  - Table area
- Every event on the channels above had information regarding the hold and finger position (if applicable)
- A discrete channel captured all events of discernible bending (regardless of player)

These 3 holds - fan, stack, and single card - do not only differ in number of cards and finger placement, I noticed that they serve different purposes. A player holds cards in a fan to scan the cards in the hand. This scanning is more complex in TCGs where each card has a lot of textual information. A simple fan will often occlude too much of that information, so I observed that some of the participating players developed a kind of slide through the cards to scan them, using the thumb to slide one card over and then another. I've observed players holding cards in a stack in order to either reorganize (straighten) the cards, or to "dump" several cards at once (this could be for good reasons such as gaining a point, or bad reasons such as discarding cards). Players also often held their "hand" as a stack while they were observing their opponent play. A player engaged with a single card for several possible game related actions: picking a card, playing a card, tapping a resource, read the information on a single card (this is common in TCGs when the players don't know their opponent's deck). All the players have frequently used card holds in both hands at the same time, i.e. while left hand hold the "hand" in a fan, the right hand draws a single card. I allowed for tagging of parallel actions.

My analysis of the observations brought to light two interesting points. One, that bends occur naturally while playing with paper cards, and those occurrences are significant in the game context. For example, images in Figure 4.2 are captured from the footage and correspond to game actions: a) shuffling the cards,

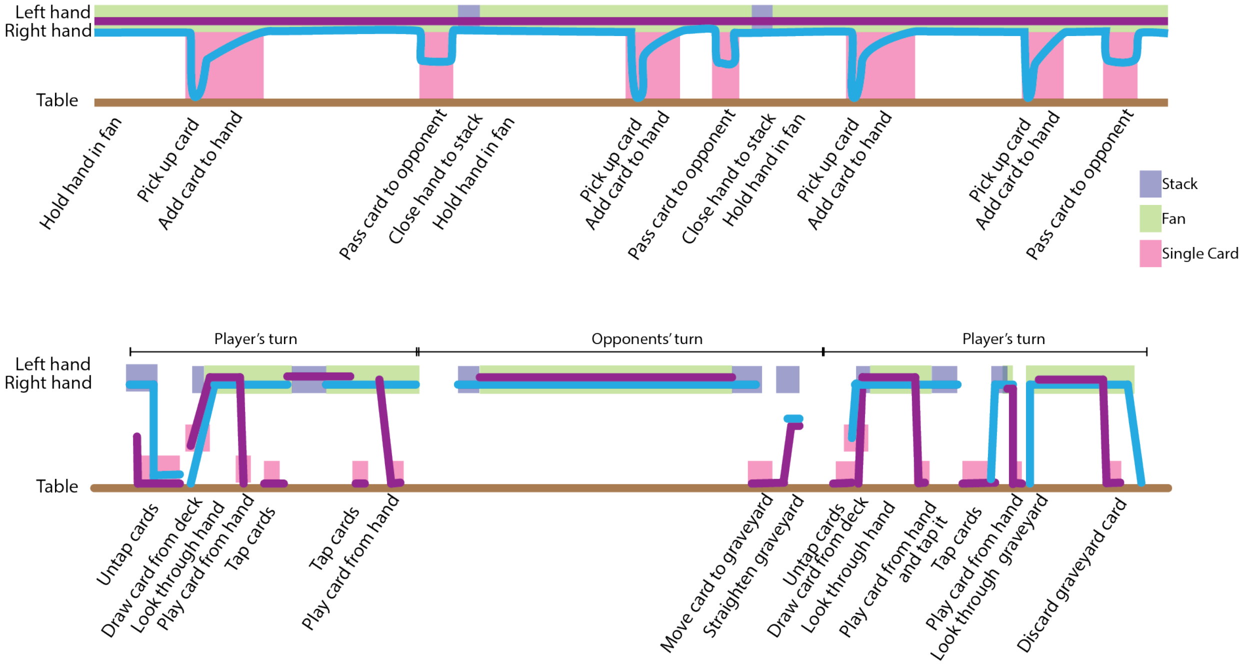


Figure 4.1: Interaction rhythm: top - from go-fish, bottom - from M:TG

b) dealing cards, c) drawing card from deck, d) position card in hand, e) picking card off edge of table, f) placing card down (playing a card), g) picking up card, and h) passing a card.

Two, my observations implied an **interaction rhythm** that is associated with a card game. This rhythm is affected by both the mechanics of a game as well as the player's unique idiosyncrasies. Figure 4.1 provides an example.

The diagram at the top shows one of the players during a session of "go-fish". In this game, the players tried to create sets of the four suits for each value, and the player with the most sets won. In each turn, a player asked their opponent for a value that they also had in their hand, *"Do you have any fives?"* , the opponent had to pass over all of their cards of that value, or if they didn't have any, tell the player to go fish for a card from the deck. When a player had a set, they placed it on the table to mark a point. This is a simple game with single cards passed either between the players or from the deck. The hand tends to become full of cards and cumbersome, which makes the player straighten and re-fan it after cards had been moved or added as seen in the figure. The purple line represents the left hand which constantly holds the "hand", the blue line represents the right hand that goes forward (in the air, or on the table) to pick up and pass cards. After the player passes the turn to their opponent, they briefly stack their "hand" to organize the cards and fan them again.

The diagram at the bottom of Figure 4.1 shows the more complex game "Magic: The gathering" (M:TG). Its pattern is more loosely defined. In M:TG, cards represented *lands*, *spells*, or *creatures*. The lands provided mana needed to cast spell and creature cards. The goal was to use your creatures and spells against your opponent to reduce their life points to zero, while the opponent used their own creatures and spells to defend themselves. As players played more cards, they built a board on the table in front of them, spreading out, yet maintaining a conventionally agreed upon layout[1]. When they used a card from their board, they *tapped* it, meaning, they turned it on its side to indicate it couldn't be used again in the current turn. At the beginning of each turn they *untapped* all the cards tapped in the previous round. The diagram shows the hectic commotion during the player's turn, where there are various untap, and tap actions that involve single cards placed on the table (mostly), along with frequent fan and stack holds of the "hand" that indicate that the player looked through their cards to decide what to do, tapped some cards to use resources, looked through their cards again to decide on another action, tapped cards to use resources and so on until the end of their turn.

4.3 Discussion

Players have their own style of playing, and sometimes luck is involved, but the rules of the game dictate a general rhythm. The possibility of this rhythm suggests that hybrid digital-physical games might be able to fit seamlessly to the way players currently play card games, by adding digital content at expected interaction points.

The M:TG rhythm in Figure 4.1 (bottom) shows how this game involved more interaction upon the table, with various tapping and un-tapping of cards, and card taps are adjacent to playing a card from hand. During the opponent's turn, the player mostly observed, but might still need to interact with their board to activate defenses, or, like in this example, remove a creature to the graveyard (it was defeated). Players can spend a great deal of time in their turn looking through the cards in their hand. This is because each card is crammed with information that is pertinent to the game, and the players must scan through the cards one by

---

[1] This individual board layout is called a tableau in card games

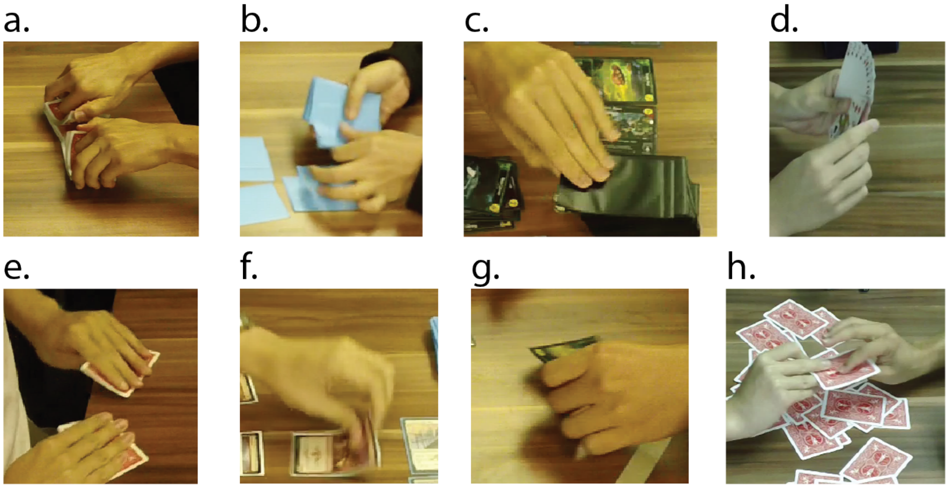


Figure 4.2: Some card playing gestures that included a bend

one sliding them over with a thumb to properly read all of that information.

It is clear from differences in the two examples in Figure 4.1 that I can't make general claims about interactions with card games. However, it does suggest that individual games can have an interaction rhythm, and if that rhythm can be identified, a system of interactive cards may be able to use it to automate, enrich, guide, or otherwise improve the game experience. It is also possible to use this rhythm to minimize the amount of cards needed in an interactive cards game, by reusing cards that serve a similar purpose or cards that were discarded. For example, in a game like M:TG, on digital card can be used as a virtual representation of all 'land' resources: when a player plays a 'land card from hand, the resource is updated in the virtual resources card and the card that was just played can be digitally reused as something else; When a 'creature' card is played from hand, the resource cost of that card is confirmed and updated in the virtual resources cards, thus automating both the tap/untap mechanic as well as enforcing mana-cost rules.

This study also helped me answer my first research question: "is bending a suitable interaction for card play?" using the sensing-based interaction framework. To match a form of interaction to an activity, the expected gestures (explored in this study) should correspond to the desired and sensed gestures. Therefore, I wanted to know if bending is expected during card play, but not overly used as to make too much 'sensed' noise. For that purpose I tagged card bends I found in the data. An example is shown in figure 4.2. According to my observations, bend gestures naturally occur in play related actions such as shuffling, drawing a card, inserting a card to the hand etc. while not occurring in resting or card inspection actions. Touch interaction, on the other hand, may cause disruption in current play habits, for example, the case of sliding the cards in hand with your thump to read through card information would trigger touch events that are not related to an actionable task. This suggested to me that bend gestures would be better positioned as "expected" gestures that overlapped with "sensed" and "desired", and might be suitable for card play.

CHAPTER 5
BEND EVENTS - STUDY AND MODEL

In this chapter I present a user study I preformed towards identifying a theoretical model of bend events. The content of this chapter are associated with the feasibility dimension of my research. The inception of this line of investigation came about in a true Research through Design moment - while developing my first prototype for the project, I became unsure about how to implement the software that makes use of bend gestures. In my experience developing software, programmers handle input using two methods: *poling* the *state* of an input device, and *listening* for *triggered* input *events* that call a handler function for that event. It was not immediately clear to me what are the states and events pertaining to bend input devices. From my perspective, as a programmer, this uncertainty had to be resolved before I could claim bending to be a feasible form of interaction. Moreover, events should be flexible enough to be useful in a variety of different use-cases.

Flexibility of event models is a key component to making a gesture commonly used; If it is easy to write an application that uses a gesture, there will be more applications that use that gesture, and the gesture will become more common. For example, touch gestures have a robust event model that supports both data-detailed low-level events and abstracted high-level events makes it easy to create touch applications, while in-air gestures are harder to interpret and harder to develop for. The flexibility of the touch event model allows developers to pick-and-choose the exact information that they need for the interaction they are implementing. If a user places a finger on one point of a tablet, drags the finger along, and then releases it - they have executed a gesture that can be used by developers in different ways: a developer for a dating application may only care if the movement of the drag was in the general direction of the left or the right, but a developer for a game with irritated birds may want to know the exact distance and angle of the movement and potentially the length of time the finger was held in place before the release.

My programmer perspective led me to research questions 4 and 5, *What is the mental model of users bending a device?*, and *How to define bend events?* These two questions join the more practical question asking how to actually make a bendable interactive card, which is the focus of the next chapter. In that chapter, I go into full detail about the process of making my first prototype, which led to this work, as well as the making of the second prototype, which is used as the apparatus for the study reported in this chapter. The following sections cover:

- a summary of bend events handling in prior work
- a study that 1) prompted participants to reflect on bend events, and 2) recorded natural bend input signals
- a synthesis of results into an *Event Model*
- implementation guidelines for this model based on natural bend signals

5.1 Background

There is no general model for bend events, however, some past works use models that are specific to their applications. Gummi [150], the pioneering work in this field, describes the first bendable prototype and provides an event model for a bendable device with five states: (1) neutral, (2) transition up, and (3) transition down events, which are continuos, (4) target up and (5) target down events which are discrete events triggered when some threshold values are reached. Other projects have also assumed a version of reading continuous events, reading discrete events above a threshold, or a combination of the two. Continuous events are often mapped to continuous actions such as zooming, and discrete events are often mapped to discrete actions such as a page flip.

To illustrate the difference in the approaches to handling bend input, let's look at a formal notation based on Card, Mackinlay, and Robertson's [26, 27] notation. In this notation, an input device is represented as a six-tuple $< M, In, S, R, Out, W >$. Where M is the manipulation operator, In is the input domain, Out denotes the output domain; S is the current *state* of the device; R is a resolution function that maps the input set to the output set, and W is the *Works* and encompass any additional information needed to explain how the input device works.

Looking at a fully continuous bend input between some angle $-\theta$ and θ, around axis Z, that is currently bent to angle β could be noted as

$$< R_z, [-\theta^\circ, \theta^\circ], \beta^\circ, Identity, [-\theta^\circ, \theta^\circ], \{\} >$$

While a discrete approach might cut-off bends above a specific value that was decided as threshold $-\theta_1^\circ$ to demarcate down events and θ_1° to demarcate up events, could look like this

$$< R_z, [-\theta^\circ, \theta^\circ], \beta^\circ, R_z : [-\theta^\circ, \theta^\circ] - f \to < UP, DOWN >, < UP, DOWN >, \{\} >$$

$$Where : f(In) = [-\theta, \theta_1) \to DOWN,$$

$$(\theta_1, \theta] \to UP$$

Though in this example, the UP/DOWN output will be repeated while β is above the threshold zone. in some implementations, such an event would only trigger once and that information would be part of the W information. In some implementations, designers chose to have 2 threshold values in order to have more flexibility with discrete events, in which case the out domain may be $< UP1, UP2, DOWN1, DOWN2 >$. And some implementations composed the continuous and discrete approaches in the same system for maximum flexibility. More details about gestures and events in prior work in Section 3.5.

Another perspective into understanding events is the division of low level and high level events. Low level events correspond directly to states of the input device, for example, a mouse button being pressed down will trigger a Mouse Down event, a mouse button being released will trigger a Mouse Up event, and a finger touching a touchpad will trigger a Touch event. High level events encapsulate more complex gestures or combinations of low level events, for example, we identify a Double Click event when mouse down and mouse up events occur twice in a short timeframe, and we identify a Pinch event when two fingers touch a touchpad and continuously move toward each other[1]. Having an event model with a wide array of events to choose from and an event system that streamlines working with events is highly beneficial for application developers. It allows for faster and more flexible code development, and in some cases, mostly for high level events, helps in defining and implementing *design guidelines* that provide a comparable experiences for users across applications.

In summary, the main objective of the study I present in this chapter is to formulate a model that encapsulates the existing approaches. Developers should be able to choose whether they want to use a discrete threshold, multiple discrete thresholds, continuous input, bend sustains, or any combination of these.

---

[1] While the notation by Card et al. supports higher level forms of input (which translate to events) with Merge, Layout, and Connect operators, I find it cumbersome; I prefer the notation for low level events.

A second objective is to provide empirical evidence that bend events should be further researched, by analyzing both users' perception on actions and effects and the difference in their natural bend movements depending on actions. Warren, Lo, Vadgama, and Girouard [171] specify a classification scheme of bend gestures based on their location on device, direction (up/down), size of bend area, angle of bend area, speed of bend, and how long it is suspended in bend (duration). Girouard et al. [52] measure durations of bend gestures in their study of one-handed bend interactions with phone sized devices to analyze the difference between the location of the bend, the hand used to preform the bend, and the direction of the bend. Neither work compared interactions in different usage contexts.

5.2 User Study

To better understand how users perception and usage of bend interfaces can inform bend events, the study participants were interviewed about the perceived timing of effects for three actions and how they reason said timing. The input signals from the measuring device were reviewed for patterns that may have significance for event evaluation. In addition, I measured some physical differences in the bends preformed during the study to get some preliminary impressions about durations and depth of bends.

5.2.1 Participants

I recruited 19 participants (6 Female) by advertising the study to CS department students and word-of-mouth recruitment. The participants had no prior familiarity with bend interfaces, and were not required to have special knowledge of interfaces or card games. They were all right handed. The sessions were conducted in a lab settings and lasted about 20 minutes each. The participants were not compensated.

5.2.2 Apparatus

To measure the bending movements of participants, I have fabricated a measuring device. The device mimics an interactive card device as I expected it to be in the PEPA project. The size of the device was 90mm × 65 mm × 7mm. It contained a flex sensor embedded in a silicone encasement, as well as two rigid boards that were placed at the top and bottom of the device to imitate the rigid electronic components (micro-controller

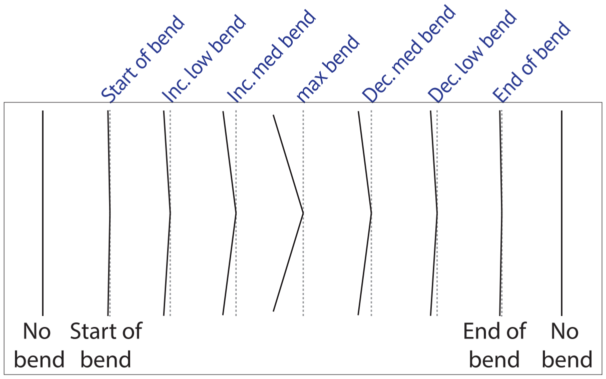


Figure 5.1: Scale of 7 bend phases

and display) in my corresponding card device. The sensors connect to an Arduino Uno controller which reads their values every 20 milliseconds and communicates via serial port with a Processing application on my computer (not facing participants). The device itself did not offer any visual feedback. The device is described as Prototype version 2 in Section 6.3.

The device is limited to 1 Degree of Freedom (1DOF) around a single, bendable, axis, to minimize confounding factors to a simple bend gesture.

5.2.3 Procedure

Each participant in the study was presented with the measuring device and was told that it will record the bending data. Participants were not instructed as to how to hold the device or preform the bends, however, I found that they had no problem bending the device correctly due to its affordance, and most held it one-handed due to its size.

During the session, the participants were presented with three tasks: (1) act as if they were using the device to view an album of images and switch to the next image with a forward bend, (2) act as if they were using the device to view an album of images and switch to the previous image with a backward bend, and (3) act as if they are playing a game on the device and an enemy spaceship is approaching and must be shot with a forward bend. The sequence of tasks was presented to participants in this order to help them become accustomed to the device incrementally (at some cost to generality). The participants did not receive visual feedback for these tasks, however, the bend signal from the device's sensor were recorded.

I chose the first two tasks since they include browsing in opposite directions which is a common action/polar-action seen in bend gesture studies. It is noted in past works [171, 52] that back bends are considers less comfortable to perform by some users, so while the purpose of the action is similar in both tasks, it is possible that they differ in users' perception. The third task was introduced as an action that may call for urgency by the users, a factor that may also affect their perception. I was looking for users' subjective opinions about the timing of the effects taking place in relation to the physical gesture.

After the participants performed the three bends, they were asked to slowly re-play the action of each task in their mind and mark on a scale of bend phases (Figure 5.1) where they believe that the "image" would change or the "bullet" would appear. They were allowed to use the device again in their re-enactment, and were encouraged, if they can, to provide a rationale for their choice.

The chart I used as a scale of bend phases showed the profile of a bend over a single gesture divided into seven minutely different bend angles: start of bend, increasing low bend, increasing medium bend, max bend, decreasing medium bend, decreasing low bend, and end of bend (in addition to the no bend, neutral position).

5.2.4 Data Collected

Data was collected in three ways: 1) the device reported bend values to the Processing software, which then saved all the read values and timestamps in a "csv" file with a participant-unique identifying number, 2) the charts shown in Figure 5.1 for the three tasks were printed and marked on paper, and 3) during sessions I took notes of participants' comments about the device, the interface, and their reflective rationales for bend phases.

I processed the generated "csv" files by hand, marking with "start bend", "max bend" and "end bend" labels using visual inspection of the signal to calculate durations and magnitude of bends.

5.2.5 Hypotheses

My initial queries into this subject showed that some users might expect that a change would take effect when the device is at maximum bend, so I anticipated that (H1) most participants will choose the maximum bend as the point of change. I also hypothesized that (H2) some of the participants will perceive that

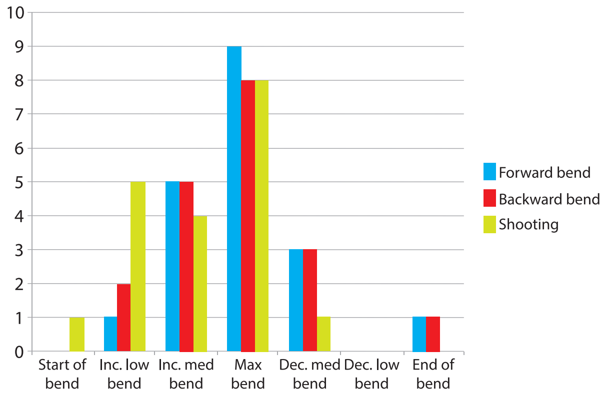


Figure 5.2: Preference of each bend phase counted across participants

shooting occurs at an earlier event phase due to its urgency. For the same reason, I thought that (H3) the magnitude of the shooting bends will be larger than the image switching bends, and (H4) their duration will be shorter, as participants may move quickly and carelessly to shoot. Finally, I hypothesized that (H5) the magnitude of the backward bend will be smaller than the forward bend, as the backward bend is physically less comfortable and participants may not be aware of the depth of their gesture, however, since the actions are perceptually the same, (H6) users will choose the same event phases for forward and backward bends.

5.3 Results

I analyzed the survey sheets from the 19 participants. Only 5 of them (26.32%) chose the same bend phase for all three scenarios, 11 (57.9%) chose the same phase for the forward and backward bends and a different phase for shooting, 1 (5.26%) chose the same phase for forward bend and shooting and a different phase for backward bend, and 2 (10.5%) had different phases for the three bends. A two-tailed Wilcoxon test of the selected phases for forward bends and shooting showed there was a significant difference ($W = 63, z = 2.18, p < 0.05$) between the chosen phases. There was no significant difference between the forward and backward bends. The occurrence count by bend phase according to the three scenarios are shown in Figure 5.2. Max bend was the most popular bend phase. Early phases were chosen for the shooting scenario more often than for the image switching scenarios, while the opposite is true for the later phases.

I graphed the output collected from the bend sensor and visually marked for each bend motion its starting

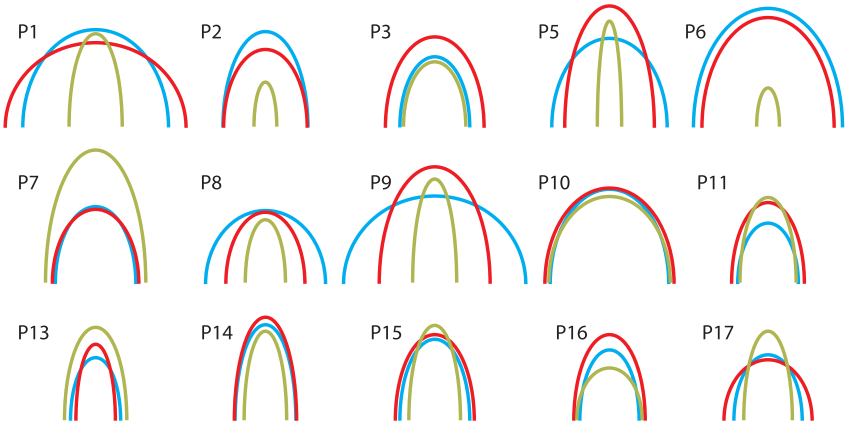


Figure 5.3: Visualization of bends performed by 15 participants

point, its peak, and its end. Unfortunately, due to hardware problems, the data from P4, P12, P18, and P19 was too distorted to be used. Some participants had made multiple successive bends for shooting (for rapid fire) in which case the first of those was measured, as that would be the first expected bullet. I calculated the duration of each gesture as its start time subtracted from its end time, and the magnitude of each gesture as the absolute value of the reading at the start subtracted from the reading at the peak.

Figure 5.3 offers a visualization of the 45 bends I manually marked with the durations and magnitudes shown in proportion. A visual inspection of this representation indicates that many participants performed a very similar gesture for the three tasks, but for some, the shooting action resulted in a shorter duration of bending. Some caveats regarding the shooting action: P7 and P16 used both hands for the gestures unlike the majority of participants, which may have slowed their movement, P10 made a prolonged gesture for shooting and explained it would fire multiple bullets while the bend is maintained, and P13 was an advocate for control-based gestures and explained that they would wait until they wanted to shoot and release to shoot.

Based on an ANOVA test on the durations and the magnitudes, the mean durations differed across conditions, $F(2,42) = 9.21, p < 0.05, MSE = 152001$. Post hoc comparisons of all pairs with Bonferroni correction for multiple comparisons showed no difference between the backward and forward durations, and significant difference between forward and shooting, $t(28) = 3.81, p < 0.05$, and between backward and shooting, $t(28) = 3.62, p < 0.05$. Changes in magnitude were not significant, $F(2,42) < 1, ns$. Figure 5.4 shows a graph of the averages.

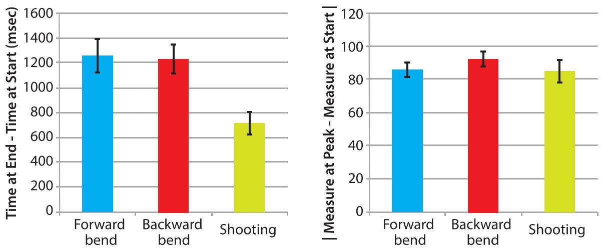


Figure 5.4: Left - averages of durations, Right - averages of magnitude

These results support some of the hypotheses, while not supporting others. H1 was supported by our survey which showed that the majority of users had some expectations about behavior at the peak. When participants were asked to explain why they chose the maximum bend, they claimed similarity to actions on other devices they are familiar with, such as, a key press, a mouse click, or shooting a gun's trigger. This is not necessary a correct mental model of these devices, however, it is how some of the participants perceived it. Only one participant suggested that image switching should occur at a later phase since it reminded them of *"releasing a mouse button"* - a more accurate understanding of mouse clicks - but that participant selected the event phase of descending medium bend and not the end of bend. One participant mentioned that the image switching actions are non-risk and reversible, so it is less important where they change. Many participants could not give a rationalization to their selection.

For the action of shooting, participants mentioned needs for *accuracy*, *responsiveness*, *fast feedback*, and *control*. Responsiveness and fast feedback were associated with changes earlier in the cycle, however, some proponents of accuracy and control had conflicting views. On one hand, it was suggested that the shooting should take effect quickly, so, as soon as the gesture starts, the bullet should appear on the screen, with the disclaimer that some low threshold is needed to avoid accidental shots that may lead to wasting ammunition or shooting allies. On the other hand, it was suggested that to maximize control and accuracy, the bullet should only appear immediately after the bend is released, which is fairly similar to using the trigger of a gun. The data shows that H2 and H4 are plausible, with shooting taking effect earlier in the bend gesture and with a shorter duration, however, there is no support for H3 as there was no significant difference in magnitudes of bend for different actions.

H5 is also unsupported by this study. There was no significant difference in the magnitudes or the

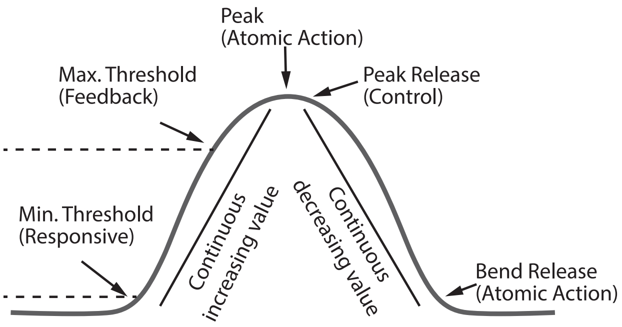


Figure 5.5: Mental model for bend events

durations of the forward and backward bends. However, in agreement with past works, some participants expressed verbally they disliked the backward bend. Four participants remarked unprompted that they found the backward bend uncomfortable, and three noted that they found it harder to execute. This difficulty may have inspired three participants to choose different phases for the forward and backward bend. The vast majority chose the same phases for forward and backward bends, as predicted in H6.

5.4 Discussion

5.4.1 Mental Model

Based on the survey results and my interviews with the participants, I formed a descriptive model of the users' mental model for the bend gesture (Figure 5.5). Users were aware that there should be a minimum threshold before an action takes place to avoid device oversensitivity. The phase immediately past this minimum threshold was deemed appropriate for actions that should be **responsive**.

Some participants chose to see effects for their action somewhere between that minimum threshold and the peak of the bend. I suggest this is related to **feedback**-based reactions - imagine the user starts bending the device and upon getting visual feedback that their action has taken effect, they realize they can now complete the gesture by releasing the bend pressure. This mode is likely to use thresholds as well, however, it differs from the previous, responsive mode by being less urgent. The thresholds triggered for feedback can vary.

The peak of the bend gesture reminded participants mouse and key clicks. I suggest that they see these clicks as atomic actions and do not pause to think about the different phases they may have (key down, key up). These participants thought of the bend gesture as an **atomic action**, executed as a phrase without cognitive pause for feedback. The model associates atomic actions with the peak, as well as with the bend release, which might be more inline with handling of other atomic actions - for example a "tap" is an event that triggers after both a "touch down" and a "touch up" events trigger.

Participants indicated that the phase immediately after the peak released was significant for actions that may require fine **control** over the exact execution time of the action, similar to the trigger of a gun.

These five points have a different contextual use case and are most likely to be useful for application developers. I took them into consideration when I designed the bend events' conceptual model, which served as the base for my software implementation for the PEPA cards.

5.4.2 Directionality

I did not instruct the participants in how to use the device and perform the bends. From past works, I expected the backward bend to be less comfortable for the users. However, I was surprised to see that 2 participants naturally performed what I thought of as the forward bend by bending *away* from them and what I thought of as the backward bend by bending *towards* them. The two specifically mentioned in their comments how uncomfortable the backward (my perceived forward) bend was to perform. This seems similar to the observation by Lo et al. [107] that some users view bends as "pushing" while others view it as "pulling". Based on this discrepancy in perceived directions, I suggest that rather than assigning directionality to the bends with names like *forward* and *backward* or *up* and *down*, we should refer to polar bend directions as *primary* direction (the "comfortable" direction) and *secondary* direction. The actual direction of the primary and secondary bends should be a preference set by the user, similar to the choice of scrolling direction in windowed systems.

5.4.3 Signal Analysis

While I was reviewing the raw input signals from the bend gestures that I collected, I noticed several patterns that show potential high-level bend events - the prolonged bend, the multi-bend, and the flick - and potential hazards for bend event identification. Figure 5.6 shows segments extracted from the raw data to demonstrate

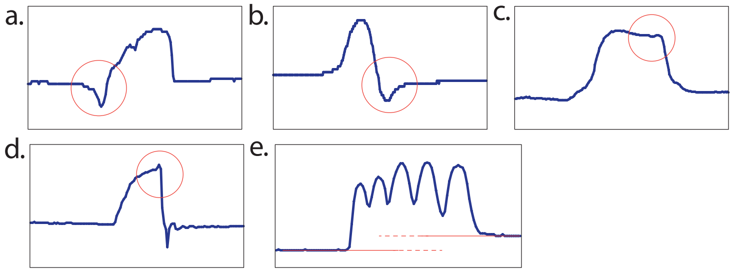


Figure 5.6: Examples of informative raw signals

my observations.

Based on the mental model from the previous section, I would need to detect low-level events for the beginning of a bend, every time a threshold value is passed, at the peak of the bend, when the peak is release, and at the end of the bend. A common way to discover peaks in signals involves calculating the derivative of the signal, smoothing it using a smoothing function, and finding where the derivative crosses the 0. This method has advantages, however, it also has disadvantages: first, depending on the smoothing function, additional signal values might be needed, which could cause a time delay in detecting the peak, second, smoothing can cause some distortion to the data, it is mostly effective against noise in the signal - but what if the signal preformed by the user is messy, not due to noise, but rather due to execution? What if the signal muddles the user's intention?

For example, Figure 5.6a shows a **wind-up**, where the user slightly bends the device in the opposite direction before their actual intended bend, and Figure 5.6b shows an **overshoot**, where the user goes over the base line when returning from a bend causing an unintentional small bend in the opposite direction. The intention of the user in both cases is to perform a single bend, they may not even notice the wind-up and overshoot, and if the system performed some action based on these unintentional gestures, the user may be confused or frustrated.

Figure 5.6c shows a **prolonged bend**, during which, the user explained, the system would fire multiple bullets. This is a reasonably expected gesture similar to tap-and-hold, however, it is difficult for users to maintain the same value for a prolonged period of time (especially on a soft device, as explained by Kildal et al. [91]), and the user's hand may cause slight peaks. It isn't the user's intention to perform multiple bends, but rather to hold a stable value, however, a system that is too sensitive to signal change might interpret this incorrectly.

Figure 5.6d, shows the signal recorded from a participant who wanted to control the release-time of the bend. This shark-tooth output indicated a **flick** - holding a bend and suddenly releasing it. This gesture suffers from large overshoot.

Figure 5.6e shows a **multi-bend**, a single gesture with multiple bends or multiple gestures of a single bend (for example, double click or two single clicks). It is interesting to note that none of the bends end at the base value, as if performed so quickly that the user could not notice or control the depth of the bends very well. Ideally, a system should be able to detect and support both interpretations. In addition, 5.6e shows that the user "shifted" the base value after completing the gesture, potentially by having a small un-noticeable pressure applied to the card at neutral state.

These examples show how smoothing alone cannot remove the messiness of the signal. I decided to incorporate several threshold parameters into a practical model for event identification. The concept of this model is bellow.

5.4.4 Practical Event Model Concept

I will now discuss the concept guiding the more practical model, illustrated in Figure 5.7. This is the base of my implementation of the event engine discussed later in Chapter 6. The Figure shows events pertaining to the *primary* direction, hence the prefix "PR_" added to event names, however, the *secondary* direction is symmetrical and would be prefixed with "SC_".

First, I divide the space of possible bend values into threshold *steps*, each of height h, with an event ("PR_THRESHOLD") triggered every time there is a change in step value. To avoid sporadic changes in step, for example, when the values read from the device oscillate between two close values that are on different steps, I added an *escape* threshold from a step, marked e_s.

As I mentioned, there are several potential problems occurring around the base value that can mislead the interpretation of the signal.To ameliorate these problems, I added a threshold for leaving/entering the *base zone*. I refer to this parameter as *escape* from step 0 threshold, and mark it with e_0. When the system would detect that the signal left the base zone, should trigger a start event ("PR_START"), and when the signal returned to the base zone, it should trigger an end event ("PR_END").

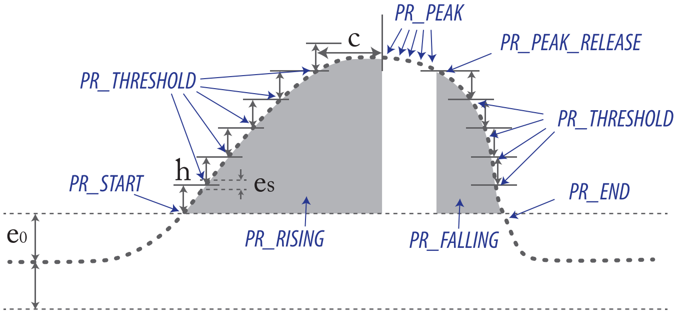


Figure 5.7: Concept for event model using threshold steps

To support systems that rely on continuous value inputs, events trigger appropriately: "PR_RISING" would be issued after PR_START and while PR_THRESHOLD is increasing, and "PR_FALLING" would be issued after PR_PEAK, before PR_END, while PR_THRESHOLD is decreasing.

At any time, the step S_t is calculated as a function of the current and last values $v_t, v_{t\text{-}1}$:

$$S_t = \begin{cases} S_{t\text{-}1}, & \text{if } |v_t - v_{t\text{-}1}| \le e_s. \\ \left\lceil \dfrac{v_t - (base + e_0)}{h} \right\rceil, & \text{otherwise.} \end{cases}$$

When the step has not changed for c rounds of input samples, the system would determine that a peak has been reached and start triggering peak events ("PR_PEAK"). A peak can be temporary, and after a small delay on a value, the user may increment the bend, thus returning to a rising mode, however, if the user starts releasing the bend, leading to a smaller step size, a peak release event ("PR_PEAK_RELEASE") is triggered. At a minimum, a valid bend cycle must have these five events trigger at least once: start, threshold, peak, peak release, and end. This means that the peak and peak release events need to be triggered even if there is no prolonged sustain of length c on any value.

This concept includes several parameters that need to be fine-tuned to make the system usable, however, its design lends itself well to a state-machine implementation, which is the simplest approach for processing input devices. More on the implementation in Section 6.3.3.

CHAPTER 6
PROTOTYPE DESIGN

An integral part in the definition of research in general, and research through design specifically, is the requirement to disseminate knowledge. There is no definitive method of describing the process of designing and building a prototype in the Research through Design field, nor is there a definitive approach to how one should represent the artifacts that are created while writing research oriented documentation. Practitioners have their own tools and methods for the design process, but they are not expected to present systematic rigor in their work. In this section, I will review several approaches for documentation and presentation of knowledge shared by design researchers. I will then describe my own approach for processing and reporting the results of my prototyping process, in this approach I focus on prototype exemplars and asses the kind of development work I preformed throughout the design, and relate them to each other on a form/function relation as a way of highlighting the priorities of the process. To do this, I will introduce the *Designer's Reframe and Refine Diagram*, a visualization tool I devised to organize my writing process. This is followed by full accounts on 5 prototype versions in the PEPA bendable interactive cards project.

The question of how to document and report RtD - and the prototypes created throughout - has been discussed by many researchers. There is one clear distinction to address: documenting is an ongoing process that must take place throughout the Research and Design processes, reporting takes the form of aggregating the documentation into a cohesive narrative that is informative to the reader and can provide them with insights; this is the knowledge dissemination portion of the RtD work and it relies on consistent documentation. The distinction between the ongoing documentation and the summative reporting can sometimes blur in the tools and visualizations presented below, however, it is commonly agreed that **documenting** the design process is an act of design in-and-of itself [9] causing a rather cyclic problem. Other problems to the documentation task involve *"determining what to document, and finding the right level of detail"* [34]. In general, design documentation is labour-intensive, and in most cases it is only possible to capture part of the process. Often, data that is gathered may be lost over time. Another inherent problem observed in design reporting stems from design decisions being interpreted retrospectively in the post-design write-up, which may skew the reports on the actual design process.

The medium used to document design is varied, covering text, images, videos and more [9]. Designers have created elaborate visualization and tools that organize and make this content available. In *'Dispelling design as the black art of HCI'* [179] Wolf et al. bring up the importance of showing the process of design in HCI settings rather than simply presenting a finished prototype (which is then usually used in a study). After explaining this argument, they show their own visualization of a project showcasing the non-linearity of the process and the different forms of judgment they employ to evaluate and drive ideas and artifacts.

Dalsgaard et al. [33] present three types of maps at three different levels of granularity to capture design: 1) An overview map meant to show the whole process in a single representation, 2) A strand map that follows the progression of a specific idea, and 3) Focal maps that spotlight specific elements within a strand. They devised iconography to represent documents on a map by medium (word, text, image, movie, model) and role (idea, inspiration, condition). The maps have different purposes in the reflection phase, for example, the items in an overview map are demarcated by "design horizon" lines that show the fluctuation in the solutions considered through divergence and convergence phases over time; The focal map poses descriptive elements (subject, approach, and outcome) against their reflective counterparts (relevance, rationale, and insights). As other works, they use a project from their own experience to demonstrate the maps and their use.

For ongoing documentation, Gaver [49] suggest **Workbooks**. A workbook is a collection of *design proposals* which can take various forms and mediums. Gaver profess that the use of a workbook helps transition from background research to the design process, can serve as a place to explore one's creativity, and a place where a designer can externalize their ideas. They share personal experience of creating workbooks especially in the early stages of design. Dalsgaard and Halskov [34] describe a tool they have made to assist with project documentation called **Project Reflection Tool (PRT)**. This tool is organized around *'design events'*, which are distinct activities such as a meeting or workshop with specific goals and a limited timescale. They describe a web-based tool they created based on this approach. The tool presents for each project a timeline where the designers can add events, sub-events (to organize within events), and notes, which are meant to help document less formal design activities. They share examples using this tool in their design projects.

Bowers [17] presents **Annotated Portfolios**. Annotated Portfolios are meant to capture the resemblances in a collection of artifacts while celebrating their differences. Bowers explains that *"annotated portfolios are proposed as a viable means for communicating design thinking in HCI in a descriptive yet generative and* *inspirational fashion"* without expectations of theory generation. Portfolios can be annotated in different ways to reflect different purposes: *"Artefacts are illuminated by annotations. Annotations are illustrated by artefacts,"* and the annotation changes how the artifacts are perceived. Bowers then describes a design project where annotated portfolio was instrumental in communicating their work, and what properties they use to form annotations.

Pictorials are another form of reporting on RtD that has recently gained popularity as a separate research track in some conferences. In *Attention To Detail: Annotations of a Design Process* [81] the authors presented for the first time a pictorial essay as a research paper in an HCI conference. The authors presented collections of images that showcase various small incremental design iterations in their design - such as trying various sizes for a funnel-like object or testing various metal finishes and their effect on reflections - accompanied by short descriptions. This approach to documentation appealed to many others who were feeling limited by textual formats. In 2014, an official track for pictorial submissions on equal footing with research papers was created at the Designing Interactive Systems (DIS) conference [14]. Other conferences have adopted the format since then.

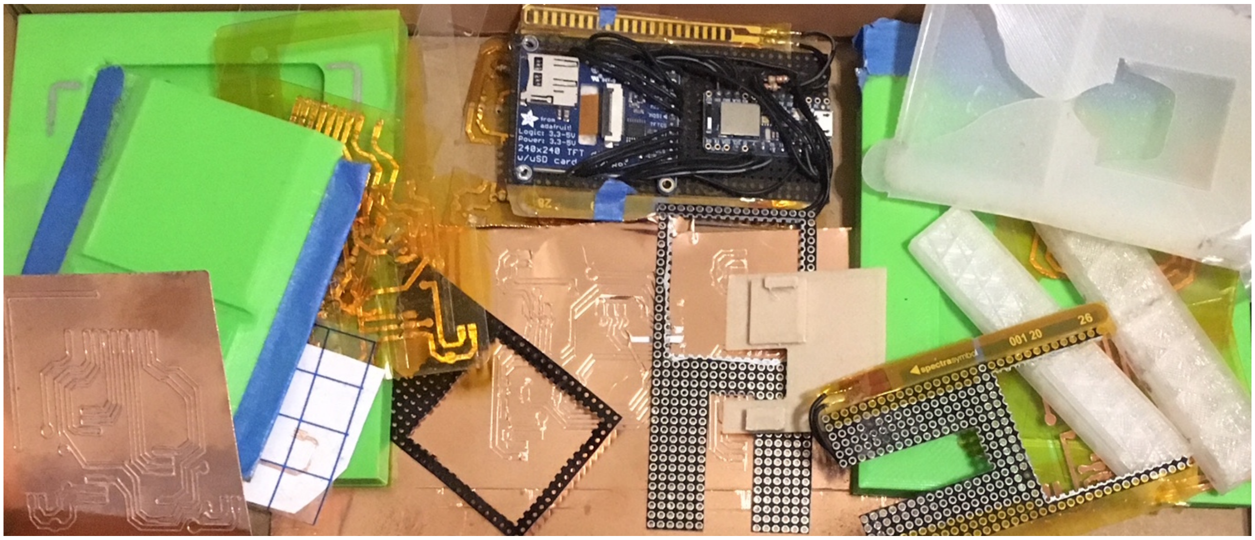


Figure 6.1: Prototyping is a messy process

One takeaway from this overview can be that documentation of the RtD process is necessary, complex, and personal. The method of documentation can differer by designer, project, phase of the project, and even specific activities. The elements that are chosen as focal points for reflection are subjective, and dependent on what was documented, how, and what is the narrative that sparks the designer's imagination for describing their work. Throughout the process, while inevitably accumulating all kinds of materials - sketches, notes, pictures, write-ups, videos, and rejected prototype - the design researcher thinks about how all the threads can come together systematically to say something new.

During my design process, I was attracted by the messy odds and bits left behind (Figure 6.1 shows some of it). To me, those pieces have their own beauty, but they are almost never part of the story, even pictorials that present iterative material experimentation [81] do so in a neat and organized way, meanwhile the messy iteration of software development is forever lost in version control and barely get any mention. At the same time, looking through other forms of summative documentation of RtD, I realized that they mostly don't fit my way of working - several of the approaches draw upon intense documentation of design activities, but these kind of activities are more appropriate in a group design effort rather than a solo design project where the line between design and implementation are blurrier than ever.

I was inspired to devise my own method for organizing and reflecting on the design process using a visualization tool called Designer's Reframe and Refine Diagram. This diagram helps me identify key moments in the process when I can say I reframed my attention, manifesting in a new distinct version of the prototype, as well as explain what was the refinement that took place while my attention was towards this prototype version. My method and the DRRD visualization tool may or may not be useful for other designers, as I say, this process is personal, however, I believe it provides a new and different approach for RtD reporting. The next section details the method and the DRRD tool.

6.1 Designer's Reframe and Refine Diagram (DRRD)

In this section present a visual annotation tool, Designer's Reframe and Refine Diagram, meant to assist in the reflection process necessary to create a methodical documentation and rationalization of a design process while focusing on key point of reframing the prototyping process and the small, iterative design through trial-and-error that leads to various design decisions. The concept revolves around the idea that prototype development is often a combination of *refinement* steps to "prefect" (I use this term loosely, prototypes are not meant to be perfect) a given prototype, and occasional idea or goal *reframing* that lead to new distinct versions of the prototype. This visual annotation tool is used to sketch a perceived progression between

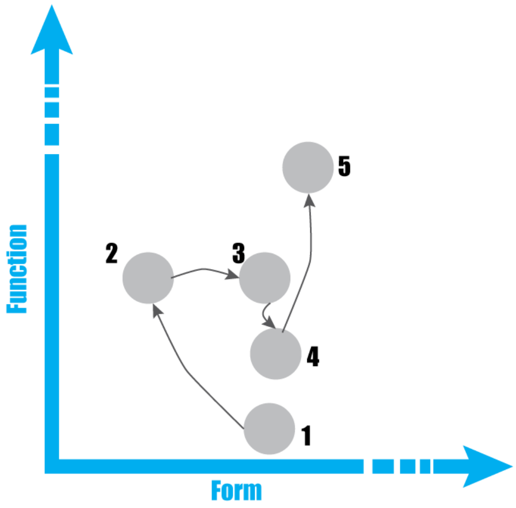


Figure 6.2: Example of the Reframe step of DRRD

versions of the prototype in a space subjectively defined by the designer of form and function, while guiding the designer to answer reflective questions about each version, answers that are then collected as the process description. First, I will describe DRRD in general, and then I will show how it is used in my project for the writing of this chapter by visualizing 5 prototype versions I've made in the form-function space.

Figure 6.2 shows an example of the reframe step of DRRD. In this step, the designers would reflect on the stages in the Research through Design process where prototypes took new directions. This is portrayed on a diagram with a **Form** axis and a **Function** axis, which broadly stand for what the prototype looks like and what it does. Form and Function are in no way mutually exclusive (for example, adding a button or changing material can affect both form and function) and there are no claims of superiority of one over the other, they are merely chosen as two aspect that can be associated with most prototypes. The designer is expected to imbue these axes with meaning - what do they interpret as part of the form in their work and what do they interpret as function. Then the designer places dots representing "reframe" prototypes in relation to each other to indicate if their form or function has become "more" or "less" and to what extent. The arrows between the version represent the kind of design and implementation work that occurred while transitioning from one version to the next.

It must be noted that a move in the "negative" direction of an axis, i.e. having "less" form or function, is **not** an indication that something is missing from a prototype - it can be a sign an intentional simplification.

For example, we can claim that the smooth glassy design of a modern smartphone has a much simpler form than an older phone with a sliding keyboard, placing the new to the left of the the old in the form axis, however, this is not a deficiency, the functionality of the new phones far exceeds that of the old one, and this functionality came hand in hand with the simpler, button-less form. Though, was this considered *incremental* ("more") or *simplification* ("less") of form and functionality can be a subjective notion that is, once again, left to the designer.

As an example, in Figure 6.2 a fictional project is represented. The designers have decided their project included 5 distinct stages of prototypes which are considered "reframes":

- First prototype version was a non-functioning version of the intended end-product which had a fairly faithful representation of the form. Such a prototype may be useful for elicitation studies or as a step towards brainstorming yet-unknown functionality.
- Second prototype was a foray into implementing some of the functionality the designers wanted to explore at the expense of maintaining the desired form. This may be appropriate as functionality is being implemented and tested and form perfection is not yet a priority for the designers.
- Third prototype includes the functionality from the second prototype, but now in a form that fits better with the original intention for the end-product.
- Fourth prototype in this example shows that designers decided to pull back some of the functionality exhibited in the third prototype. This could be, for example, a decision based on the results of a user study showing that some of the functionality was frustrating or misunderstood.
- Fifth prototype shows a large increase in function and some additional change to the form. This could be a later prototype that was created to incorporate new features, while using the fourth prototype as a starting point for design.

Following the "Reframe" phase of this reflective process where distinct versions are defined, we enter the "Refine" phase. In this phase we consider all the small iterative changes that were done on a prototype within the specific prototype version. More often than not, these changes can be quite small and occur over many iterations making it unrealistic to document all of them in a digestible research-article format. The refinement annotation is meant to help make a symbolic notation of the general direction of changes made while working with a specific version in both form and function. These symbolic notations are sketched inside the dots from the "Reframe" phase. Figure 6.3 shows the options for these symbols.

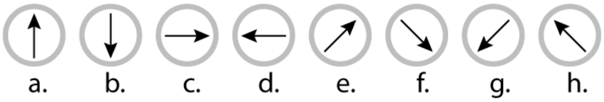


Figure 6.3: Refinement iconography

The symbols shown in Figure 6.3 represent: a- incremental addition of functionality, b- incremental simplification of functionality, c- incremental addition to form, d- incremental simplification to form, e- incremental additions to both form and functionality, f- incremental additions to form with incremental simplifications to function, g- incremental simplifications for both form and function, and h- incremental simplification of form with incremental addition to functionality.

6.1.1 DRRD for PEPA cards

Figure 6.4 shows my Designer Reframe and Refine Diagram for the PEPA project that I used to guide the process of writing this chapter. I decided to divide my prototyping process to 5 distinct "Reframe" steps marked 1 to 5 (representative versions from each prototype version are shown in Figures 6.5, 6.8, 6.11, 6.12, and).

The first prototype was a proof of concept version of interactive cards, it had very little of the form I wanted but had a functioning version of a bendable game. While working on this prototype, I mostly focused on writing and improving the code running the cards, so I chose to use symbol (a) to show this iterative process.

The second prototype was a tool I used to measure bend gestures in a study. It only contained bend sensors without a display, so I consider it less functional, but in form it was closer to the shape and feel I wanted for a bendable card. I once again use the (a) symbol to show the incremental progression with this prototype, this may seem counter-intuitive since I place it as having less functionality than prototype 1, however, I used this prototype to develop a lot of the event engine for the cards (which will be descried later in the chapter) so, overall, I think of this prototype as a prototype where my work involved incremental

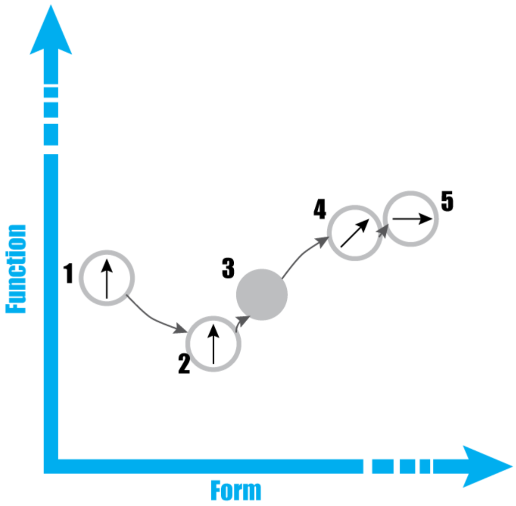


Figure 6.4: Applying the DRRD to the PEPA prototyping process

additions to functionality.

The third prototype was my first attempt to build a complete card version, it was based on prototype 2 but with the addition of a display. With the display, it was slightly closer to the form I wanted, and since, technically, the display worked properly, it had slightly more functionality than prototype 2. However, I immediately knew this prototype is not what I wanted - it was not soldered, so it used bulky components and jump wires - so I discarded it. This prototype had no symbol for the incremental steps to highlight how quickly it was discarded.

The fourth prototype is a far better version of prototype 3, this time with components and wires properly soldered. I spent a long time in this Reframe step working on improving both form and functionality, so I represent the steps with symbol (e).

The fifth prototype is a direct continuation of prototype 4, however, I distinguish between them because they use different materials, which greatly affects and improves the form. For the same reason, I used symbol (c) to represent the refinement process. While making the decision to separate this version from prototype 4, I had to properly reflect about the reasons I was not satisfied with 4.

The next section outlines the full process.

DRRD Reflection Method

1. **Review** - Reflect on all of the prototyping process from the very beginning of the project including all successful and failed versions.
2. **Reframe** - Define a partition into versions - stages in the process that you perceive as being distinct from each other, where you had to reframe your approach to the design. Each version should have at least one representative depiction, a sketch, photograph, or screenshot, that were captured during the ongoing documentation phase.
3. **Transitions** - For each version ask yourself:
   (a) what was the reason to create this prototype version, what was its goal?
   (b) is this prototype version based on a previous prototype version?
   (c) compared to the previous prototype version, does this prototype version have a simpler or more complex form?
   (d) compared to the previous prototype version, does this prototype version have a simpler or more complex functionality?
   (e) how did you think that the changes for this prototype version, compared to the previous one, will help achieve its goal?
   (f) in what way did the changes for this prototype version, compared to the previous one, help or hinder achieving the goal for this version?
4. **Refine** - For each version ask yourself:
   (a) since a recognizable instance of this prototype version emerged (the representative depiction may be a good starting point), what type of iterative changes did you mostly implement? to form? to function? adding or simplifying?
   (b) what made those types of refinements compatible with this prototype version?

   (c) was the refinements aligned with the goal of the prototype version? if not, what inspired the refinements?
   (d) while working on these refinements did you discover strong advantages or drawback in this version's approach that inspired moving to a new prototype version?
   (e) while working on these refinements did you come up with novel ideas that inspired moving to a new prototype version?

5. **Sketch** - Sketch the DRRD to illustrate your thoughts.

6. **Write** - Write down a detailed description of your designs and rationalizations - For each version write:

   (a) a full description of that version (all points relating to form and function).
   (b) the goal of that prototype version, how you though to achieve it and whether or not you achieved it.
   (c) what kind of iterative development you worked on using this prototype and why.
   (d) what conclusions you had from your work on this prototype version and how it inspired the changes you implemented for the next version (if there is one).

6.2 Prototype 1 - Proof of Concept

6.2.1 Prototype Description

For the first prototype, I built two interactive card units and tested them with a simple game called "Spider Catch". For each card I used off-the-shelf components (purchased from Adafruit[1] where I also bought all of the components for other prototype versions): an Arduino Uno board, 2 Nokia 5110 LCD screens, a Spectra Symbolflex Sensor 2.2", and an nRF24L01 Wireless Transceiver. The connections are shown in the fritzing[2]

---

[1] https://www.adafruit.com/

[2] https://fritzing.org/

diagram 6.5 (on the left). All of the components were assembled in a typical prototyping environment using breadboards and jumper-wires to hold the components together without soldering. To get a semblance of the card shape, the screens and bend sensor were taped to a foam surface (See Figure 6.5 center and right).

Figure 6.5: Prototype 1 - Proof of Concept

The prototype ran a simple game called "Spider Catch". The software was developed using the Arduino IDE and uploaded directly to the Arduino Uno boards. The game engaged both cards. A spider would "jump" around and appear at random locations on one of the four screens (top and bottom screens of card1 and card2), and stay visible for a brief time. The player had to "catch" the spider by bending the correct card to get a point. If the player delays for too long after the appearance of the spider, or if they bend the wrong card, they are not given a point and the spider reappears elsewhere. The game would run for 20 rounds and the player would aim for the highest score.

6.2.2 Prototype Goal

In my approach to the design of this first prototype, I thought about the properties of cards.

- They are about palm sized (players hold them comfortably in their hands)
- They are thin (players can hold a few cards together with no difficulty)
- They are durable (players can play without worrying about damage to the cards in most reasonable uses)
- They are flexible (this is due to the nature of paper products, players use this to shuffle and peak at cards, and this quality contributes to their durability)

Therefore, my concept of digital cards embraces all of these properties in addition to including a computer in each card that would run it and automate a card game, add rich media, randomize, and preform other game mechanics unique to augmented games (see the hybrid digital boardgame model from Chapter 2 for a comprehensive list of possible digital contributions to traditional boardgames.) One of the goals of the cards is to maintain *awareness of game state*. This awareness of the game state, dynamically influences the information that the card presents to the player. To allow the cards to keep track of the game state, the player must preform actions that indicate what they want to do. In this project, I was inspired to use bending as the form of input, i.e. when players preform actions, they bend the card or cards and this would bring about change in the game state.

In this version of the prototype, I wanted to put these conceptual thoughts about interactive cards through a reality check - I wanted to see if the basic idea has any merit, and to do that I decided to implement a "bendable" game between two digital cards. I have not worked with bend sensor or display components before, so learning how to connect these components and write code for them was an implicit goal for this version. But, again, the more explicit goal was to see if I can make it happen - build the cards, write the game. This version of the prototype was successful at achieving both implicit and explicit goals. In addition, the challenges I encountered during the development of this version, helped form future direction of study and further development.

This prototype was also built as part of project requirements for the class EE626: *"Rapid Prototyping of Electrophysical Devices"* and therefore it was created during the limited time frame of a single semester.

6.2.3 Prototype Refinement

The assembly of the cards was straightforward based on connection diagrams available online and using the solder-free prototyping breadboard and jumper wires, so the prototype refinement period was dedicated to implementing the game software. The code for the game was written in the Arduino IDE using publicly available libraries, such as Adafruit-GFX-Library[3] used for drawing on displays from microcontrollers.

The first design question I had regarding the software was: *How do the players indicate that the game should start?* What is the metaphorical "on" button for the game? It seemed that players would need to use

---

[3] https://github.com/adafruit/Adafruit-GFX-Library

a unique gesture to say that they are ready to play.With only bending available as a form of input, I decided that bending both cards together would be the best signal to start the game as 1) it indicates that the player is holding both cards, and 2) it is less likely to happen accidentally (compared to bending one of the cards). After this initiating double-card bending gesture, the cards should switch into game-mode where the cards need to be bent one at a time to catch the spider.

My first implementation of the software was naive. When I first started designing the software, I wanted each card to be an independent agent (hence the name Paper-like Entertainment Platform Agents) in an autonomous multi-agent system. That would mean that each card, would use their own sensor and the data communicated from the other card to locally determine the global state of the game and act accordingly. I generated in each card independently a data structure that represented the 20 rounds of play, and relied on each card to correctly maintain game-state based on the current round and the bend values they self-measure and receive from the other card. Figure 6.6 shows the state diagram of my code for this approach. However, after implementing this scheme, it was evident that the cards would not synchronize properly.

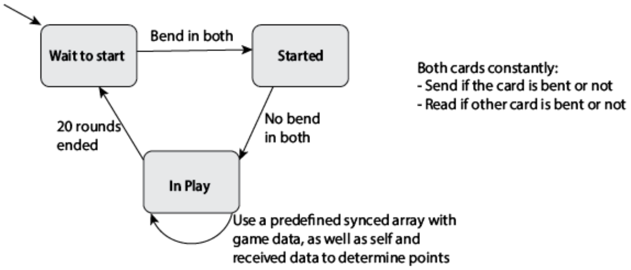


Figure 6.6: State machine of naive implementation

Rather than trying to fix synchronization problems in my symmetrical solution, I changed my approach to a Master-Slave design. Figure 6.7 shows the state diagrams used by the slave card and the master card according to this scheme. In this implementation, the Master card was in charge of constant synchronization between the two cards, informing the other card which state to switch to and then waiting for an acknowledgement before progressing its own state. Both cards still generated their own data structure with information about the 20 upcoming rounds. The slave card would persistently transmit its current state and whether or not it is bent. The master would read the values, evaluate the new state (if any), and send one of 4 possible messages: Wait_msg, Start_msg, Play_msg followed by the round index, and Sync_msg followed by the score value. This version of the code worked correctly, however, it was a rather convoluted way of running a very simple game and was not going to scale to more games with more cards.

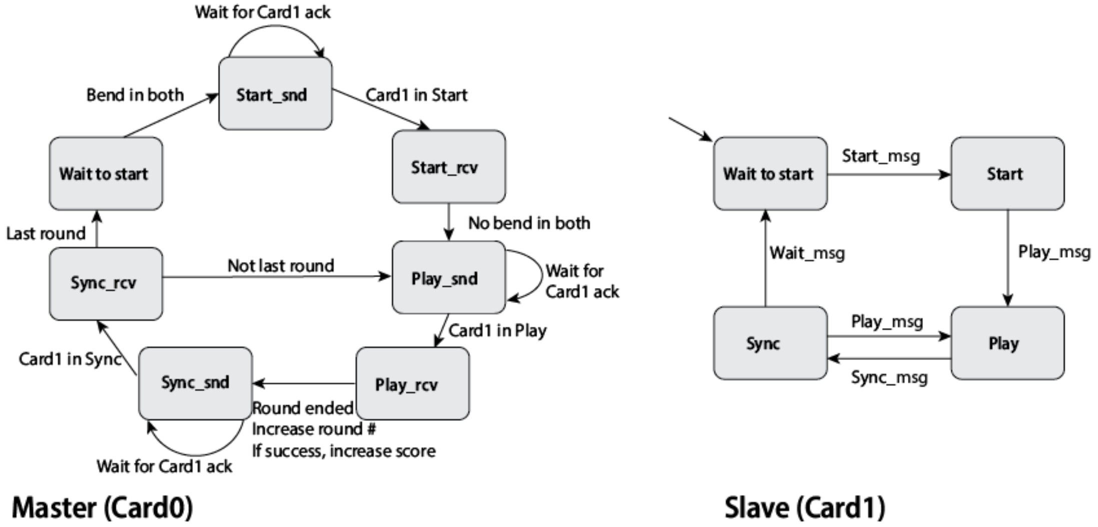


Figure 6.7: State machine of Master/Slave implementation

6.2.4 Lessons Learned from Prototype

First, it was clear that the code for the single game I created in this prototype version was too complicated and unscalable. Moving forward, I decided that it would be easier to generalize the system, making it managed by a single central control unit. This controller would handle maintaining the system, orchestrating communication, and managing all of the game logic. One of the main reasons I made this decision was the impracticality of managing game logic on-card - for this single game, the game data I generated and rules programmed within the Arduino micro-controller code, used about half of the available memory. The memory of most micro-controllers is too limited and not likely to support multiple games and it would not make sense to change the card firmware for each game. However, this issue is solved with a permanent, passive code on the cards side. This code should only have three tasks 1) establish connection to the controller, 2) continuously inform the central controller the bend status, and 3) continuously wait for instructions on what to render on the displays.

Second, the use of network transceiver was adding overhead on an already-tricky-to-synchronize situation. Since there are still many unanswered questions regarding the bending interaction, I decided to refocus my project exclusively on the interaction aspects of the device and leave network management to a future project. For that reason, future prototype card versions were connected to a computer as a central controller and communicated via the computer's serial port, which is easy to establish using the Processing programming environment[4].

Third, it was not clear what is considered *to be in a bent mode*. When I started the implementation, I hard-coded a value in the code and any sensor value above that number indicated that a card is *bent*. This is a reasonable approach, however, it felt arbitrary to me. There seemed to be many phases throughout the movement needed to bend a card, so when would be the best time to trigger a change in my state machine? Maybe when I just start the bend? maybe when I complete the bend and the card is neutral again? maybe the hard coded value is fine? I realized there is a lot I don't know about the bend gesture. Following this prototype, I decided to conduct an exploratory study (described before), and created prototype 2 as a measuring tool.

6.3 Prototype 2 - Measuring and More

6.3.1 Prototype Description

The device I created as my second prototype mimics an interactive card device in its form, though only containing sensors. The size of the device was 90mm × 65 mm × 7mm. It contained a flex sensor embedded in a silicone encasement, as well as two rigid boards that were placed at the top and bottom of the device to imitate the rigid electronic components (micro-controller and display) in, what would be, the corresponding card device. The device also included accelerometers attached to each of the two rigid boards. The sensors connected to an Arduino Uno controller which read their values every 20 milliseconds and communicated via serial port with a Processing application. The device itself did not offer any visual feedback. Once again, I have used breadboard and jumper-wires, however, the wires were extended to such length that the device could be manipulated freely at a distance from the micro-controller and breadboard.

There were two versions for the software driving this prototype, one that was created for the study,

---

[4] https://processing.org/

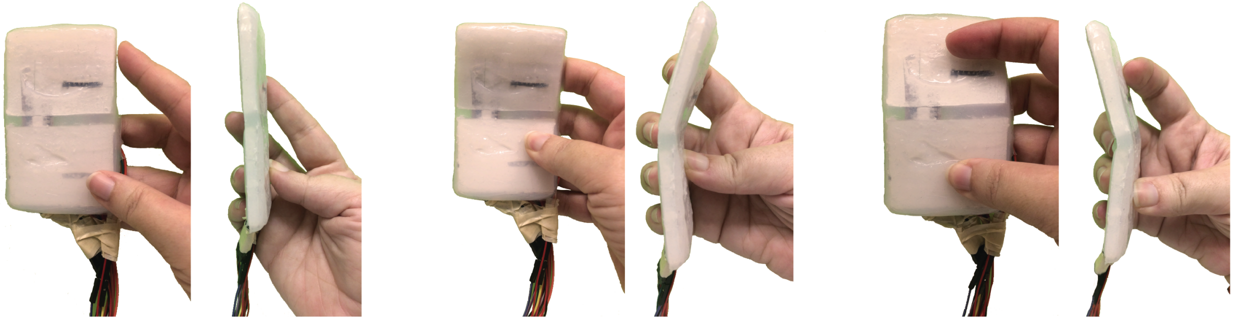
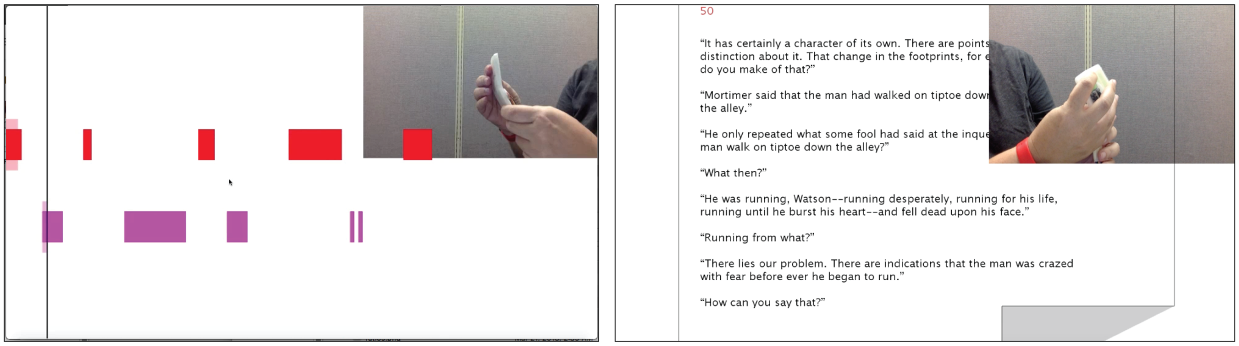


Figure 6.8: Prototype 2 - Measuring and More

and the other was created after the study, both were written in Processing3. The study software received messages with a timestamp and the values read from all the sensors from the device via a serial port. This message was wrapped with "<" ">" characters to make parsing easier. The timestamp and sensor data was continuously saved to a ".csv" file on the local file system. At the same time, a visual representation of the bend signal (from the single bend sensor) was graphed on the application window. This helped me observe and troubleshoot during the study as I asked participants to bend the device.

The second software version was implemented after I developed the event model described in Chapter 5, and was subsequently adopted as the "event manager" in future prototype versions. The event manager received the current value of the bend sensor and using a state machine (see more details in the Refinement section) called functions corresponding to the events: bend start, bend threshold, bend peak, bend peak release, and bend end. This code was reused in several applications showcasing how the event detection works. The visual part of these applications was rendered on the computer running the Processing code, since this prototype did not include its own display. Figure 6.9 (on the right) shows, as an example, an e-reader app, where a small bend forward or backward switched to the next or previous page respectively, and a bigger bend (based on threshold bend events) caused a 20 page jump in the respective direction. Another example shown in Figure 6.9 (on the left) is of a rhythm game that used the bend start and bend end events to demarcate engagement with a random-length generated "beat".



50

“It has certainly a character of its own. There are points
distinction about it. That change in the footprints, for e
do you make of that?”

“Mortimer said that the man had walked on tiptoe dow
the alley.”

“He only repeated what some fool had said at the inque
man walk on tiptoe down the alley?”

“What then?”

“He was running, Watson--running desperately, running for his life, running until he burst his heart--and fell dead upon his face.”

“Running from what?”

“There lies our problem. There are indications that the man was crazed with fear before ever he began to run.”

“How can you say that?”

Figure 6.9: A rhythm game and an e-reader app with bend interactions

6.3.2 Prototype Goal

This version of the prototype was created first and foremost as a tool for a user study. Part of the study was to observe the signals different people produce with naturalistic bends, so I wanted this prototype to "feel" similar to the final product (or at least, the way I thought at the time the final product would feel). The form of the device drew inspiration from the work of Lee et al. [103], which suggested that users preferred smaller devices, and the work of Kildal et al. [91] , which showed that users preferred devices that are softer and easy to bend. The idea to encase the cards with silicone was inspired by prior work [106] describing the fabrication of a bendable device.

The prototype was successful at accomplishing its goal - though there were occasional problems with the system during the study[5], they were easy to resolve, and the device did not break or malfunction even after the study was completed. Since it remained in good working order, I used it for the secondary purpose outlined in this section of implementing the event manager.

Based on my study and met-analysis of prior work, I formulated a model that encapsulates the existing approaches to bend events so that developers could choose whether they want to use a discrete threshold, multiple discrete thresholds, continuous input, bend sustains, or any combination of these. Following the study, I extended my goals for this version of the prototype to include showcasing how the event model can be used to fit different developer needs. This was achieved with the implementation of the event manager and example applications.

---

[5] Due to technical difficulties I had to discard the raw signal files from 4 study participants.

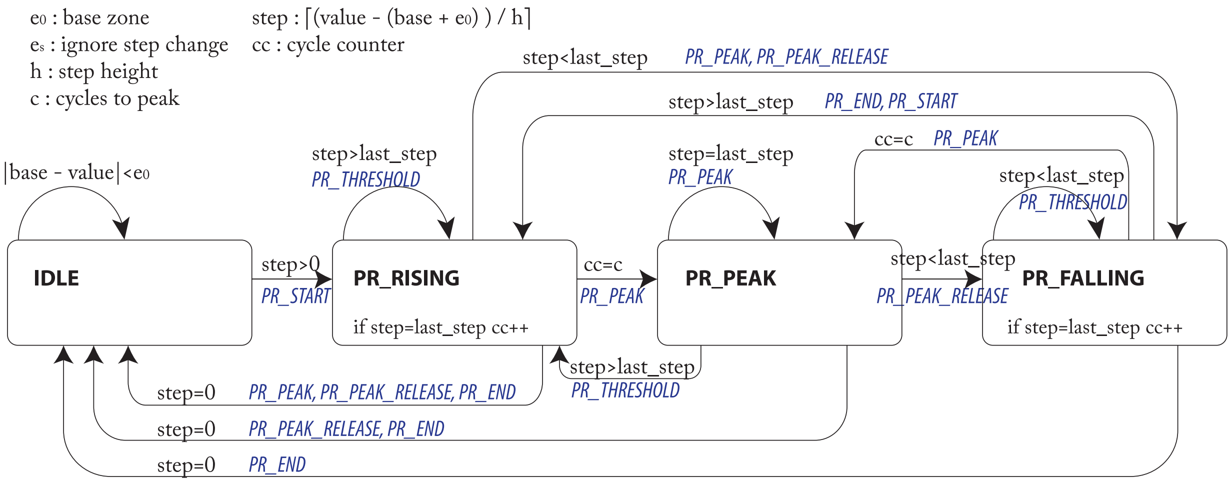


Figure 6.10: State machine for identifying bend events

6.3.3 Prototype Refinement

The bulk of refinement work in this phase revolved around implementing and testing the event model developed following the bend-phases perception user study. According to the event model, in addition to continuously supplying the raw input readings from the sensor, the system must also provide events for the start of the bend, peak of the bend, release of the peak, and release of the bend, as well threshold based events that trigger when the bend value increased or decreased by a certain amount (threshold events are meant to be a more reliable way of tracking signal increase/decrease since they are more robust against noise).

The concept guiding the implementation was illustrated in a previous chapter (Figure 5.7). The space of possible bend values was divided into steps of height h, and an escape threshold from a step, e_s, to minimize oscillating between steps due to noise. A base zone was defined as the static value that indicates that the device is in neutral state (no bend), and another parameter e_0, i.e. escape threshold from step 0, was used to provide a thicker threshold band around the base to avoid various signal confounders like overshoots and wind-ups. When the step had not changed for c number of input samples, the system would determine that a peak has been reached. Even if a step value had not been stable for c turns, the system would force a peak and peak release events before the bend end event could trigger. At any time, the step S_t is calculated as a function of the current and last values $v_t, v_{t\text{-}1}$:

$$S_t = \begin{cases} S_{t\text{-}1}, & \text{if } |v_t - v_{t\text{-}1}| \le e_s. \\ \left\lceil \frac{v_t - (base + e_0)}{h} \right\rceil, & \text{otherwise.} \end{cases}$$

A state machine is the ideal tool to ensure that bend cycles are valid. Figure 6.10 shows the state machine that produces the events as described above. (The figure shows events and states pertaining to the primary direction, however, the secondary direction is symmetrical.) The state machine has 4 states: Idle, PR_Rising, PR_Peak, and PR_Falling. In every round, 1) the value of S_t is calculated and compared to S_{t-1}, 2) when applicable, a cycle counter (cc) is reset, incremented, or compared to c and, 3) the transition rules are resolved to determine to next state.

In addition, I implemented an *event object* that holds the direction of the bend, the time it was detected, its raw value, and the name of the event triggered, which is sent along with the triggering of the event.

Here is an example of using the event object to extract the step value. This code was taken from the e-reader example application where the step value was used to decide whether to skip one page or 20:

```
void onPrPeakRelease(BendEvent event) {
  if (event.step_value <= 2) {
    prev_page(); // function to skip 1 page
  }
  if (event.step_value >= 3) {
    prev_section(); // function to skip 20 pages
  }
}
```

The event model depends on the four parameters h, c, e_0, e_s, so part of this refinement process involved exploring these parameters. I used the 15 data files collected during the study to evaluate results.

The escape from base parameter, for example, is particularly important. Using my data, which contained 93 distinctive peaks, I ran the event detection program with $h = 13, c = 4, e_s = 4$ and compared the e_0 values of 35 and 45[6]. With $e_0 = 35$, I found 19 errors in identification of bend start events, however, all errors were caused by fairly noisy data which triggered false positives. All 93 gestures where found with an average delay of 189.2 milliseconds from their apparent start to the firing of the start event. With $e_0 = 45$, I found 20 errors in identification of bend start events, however, these errors were more severe, a similar false positive issue occurred in addition to 4 false negatives. For the 89 identified gestures, I found an average delay of 224.3 milliseconds from their apparent start to the firing of the start event.

This initial exploration shows there is a fine balance to be found. Making the base zone too small won't stop the recognition of wind-ups and overshoots as events, however, making the base zone too large can cause other problems and further delay identification of bend start eventsIn addition, changing the step size can greatly affect the number of the more continuous events triggered. When I ran the program with $e_0 = 35, c = 4, e_s = 4$ and compared h values of 13 and 7, for $h = 7$, the amount of peak events triggered was 94% of its $h = 13$ parallel, while the amount of threshold events triggered was 140% of its $h = 13$ parallel. And so, while the event detections system was working reasonably well, it was far from perfect, and there may be better approaches to implement the event model (discussed in Chapter 8: Future Directions).

6.3.4 Lessons Learned from Prototype

I learned many things while working on this prototype version, both while asking my study participants to bend the device, and while writing code to implement the event manager.

First, it was difficult for participants to imagine the device as a card, because it wasn't thin enough and it had no visual component. I hoped that having no visual feedback would help remove bias from the participants who were asked to visualize in their mind the changes that would take place in three scenarios, and perhaps it had, but it also made the device unrecognizable as a card. On the other end, the silicone shell around the card was easy to bend and was not damaged through repeated use, making it a promising material.

The event model was implemented and, after making small experimental applications, showed me that the model was flexible enough to support different developer needs, as I hoped it would. It is far from a perfect implementation: it is not clear how well people would be able to perceive the differences between

---

[6] It should be noted that while the possible values for an analog sensor, like the bend sensor, measured on Arduino is in the range 0-1024 (reflects the voltage through the sensor while it is bent), the actual values I encountered were usually in the 600-900 range.

the different events (if at all), and the accuracy of the system was dependent on several adjustable parameters. However, I decided this system is sufficient for identifying events for the purpose of my project as my overall goal was to create a minimal viable version of a bendable card, and rather than continuing the work on the event manager, I opted to move on and work on adding the display component.

6.4 Prototype 3 - A First Step

6.4.1 Prototype Description

The device developed as the third version of the prototype is shown in Figure 6.11. It uses a 1.44 inch TFT LCD display, a small micro-controller with Arduino capabilities (Adafruit Feather nRF52 Bluefruit LE), and a bend sensor encased in a silicone casing. The electronic boards were placed at the top and bottom areas of the card, allowing for the center to bend forwards and backwards (center and right images of Figure 6.11 respectively). At this point I was hesitant to solder component directly on top of boards, so I tried to continue using prototyping methods - while I discarded the breadboard and got a physically smaller micro-controller, I still used headers and jumper-wires to connect component resulting in the disappointing form factor. While the silicone material remained pleasant to hold, this device was twice the thickness of prototype 2 making it harder to bend. The size of the device makes it look more like a personal game-console (like Game Boy[7]) than a card device.

6.4.2 Prototype Goal

The goal of this prototype was to incorporate an on-card display and micro-controller into the design of the previous version. My approach for this build was similar to my approach for prototype 2, to a fault - I continued using headers on the components and jump-wires to connect them, which is ideal for dynamic prototyping, but create bulky contraptions that are counterproductive to minimizing the form. So, while this version had a working display, it was not good for anything beyond achieving the goal superficially, since it was twice as thick as the previous version and unwieldily.

---

[7] A console by Nintendo popular during the 90's.

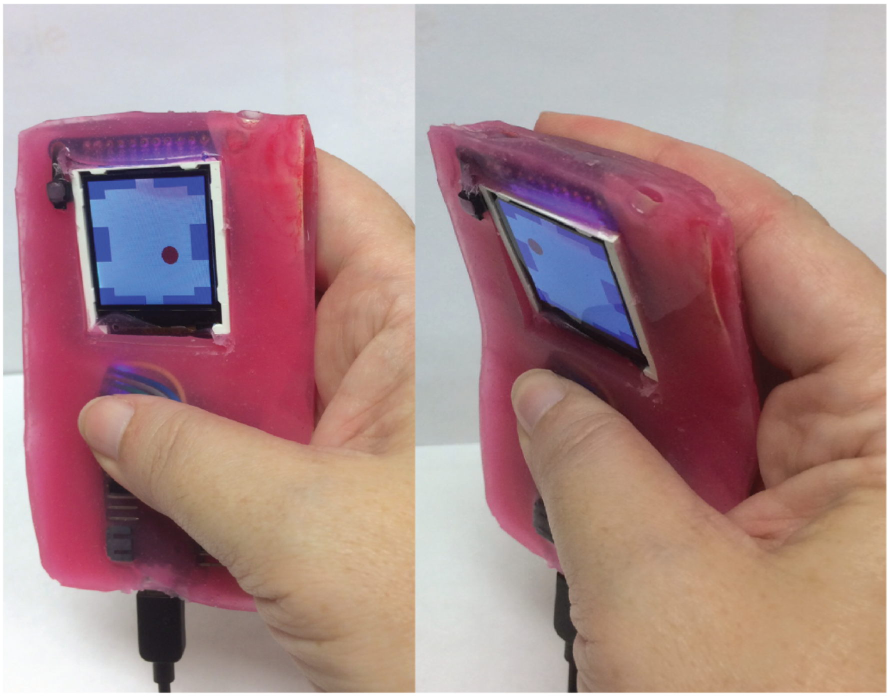


Figure 6.11: Prototype 3 - A First Step

6.4.3 Prototype Refinement

This version was deemed unacceptable, and was discarded in favor of the next prototype.

6.4.4 Lessons Learned from Prototype

This version along the prototyping path was instrumental in learning what doesn't work. I consider it as a separate version, since it was a complete and functioning device differing from the prototypes preceding it and those that came after. It serves as a turning point between prototyping the electrical components in a temporary way, and committing to soldering the components directly to each other. It would be nearly impossible to minimize the device's thickness without changing the approach. It also became clear in this version that the device needs some kind of surface to provided backing for the other components (this was not the case in prototype 2) and to ensure the encasement process is successful. In addition, given that I resolved to use serial communication to the computer, the cable connector for the micro-controller needs to be positioned along the edge of the card, and while I considered positioning the micro-controller sideways, it would not be comfortable to grasp the card in that orientation (with the cable connector protruding from the side of the card). So, it was determined that any on-card micro-controller needs to be orientated so that the cable connector is at the bottom of the card. This decision posed some limitations toward future selection of micro-controllers and possible wiring.

Finally, in this brief experience writing code for the display, I learned that rendering to the screen can be quite involved. In order to minimize the size of the firmware code on each card (due to small memory size on micro-controllers) I decided to restructure my central controller software to work with the increasing complexity.

6.5 Prototype 4 - Looking Better

6.5.1 Prototype Description

This version of the prototype assembled all the components on top of a flexible proto-board on which I soldered the components and their connecting wires. In this design I used the "Adafruit ItsyBitsy nRF52840 Express - Bluetooth LE" micro-controller (making it ready for future project versions that incorporate the network engine), "Adafruit 1.54" 240x240 Wide Angle TFT LCD Display with MicroSD", and 2 bend sensors (1 facing forward and 1 facing backward). I have tried many types of wires to use for connecting the components, and silicone wrapped wires worked best for me. For this prototype, I designed and printed 3D molds for casting the silicone for the cards and tried 3D printing internal rigid parts to compensate for the unevenness of components. Figure 6.12 shows exemplars of cards from this prototyping step.

Aspects of the software from Prototype 2 and 3 were carried over to this prototype, but the architecture of the software became better defined and a graphics engine was implemented on the Processing side of the code, and appropriately, a loop to read incoming graphics messages was added to the firmware of the cards. More details in the Refinement section.

6.5.2 Prototype Goal

This prototype's goal was to be the final outcome of my dissertation project. It was supposed to be a card that incorporates the on-card components in a thinner form factor (compared to prototype 3) and be used to

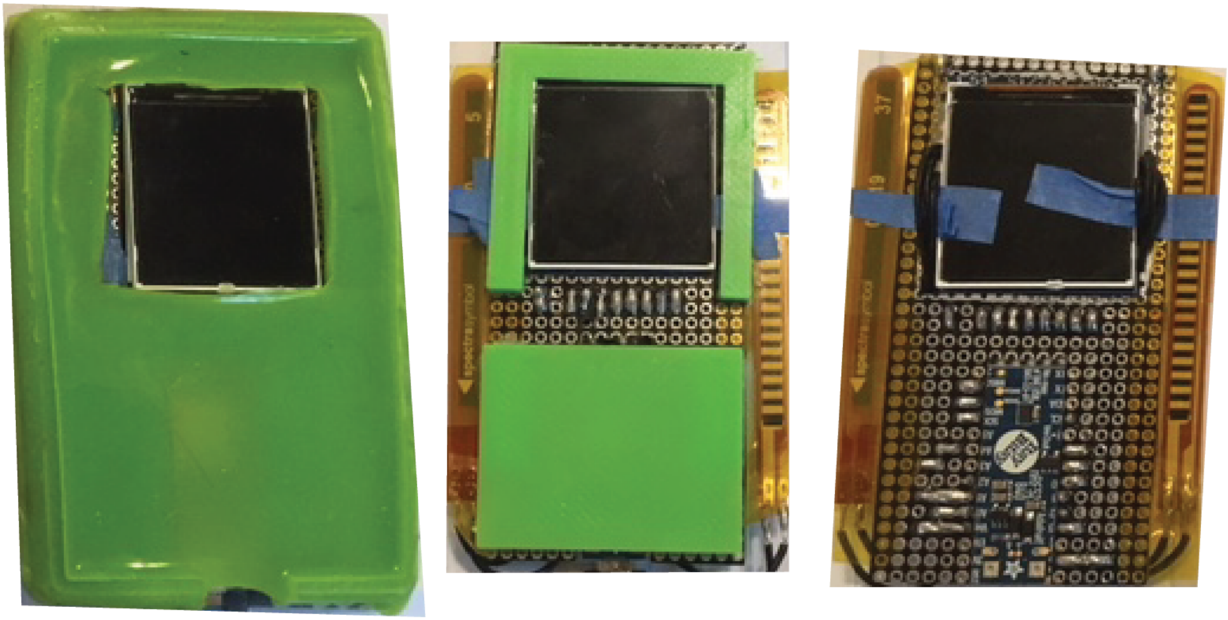


Figure 6.12: Prototype 4 - Looking Better

showcase the project and perhaps be used as a tool for studies.

The sharp reader would have already noticed that this prototype version failed its goal as it is not the last prototype in this chapter. The cards never felt "right": they were still too thick; the rigid components did not fit naturally to make the surface feel smooth; when the silicone was poured directly on the components it would "yank" on the wires causing occasional disconnects; when it was poured as a separate sleeve, it puckered when it was bent and still the connections were not stable enough. In addition, I had to cut holes in the flexible proto-board I was using as the surface base of the card, which made it brittle and caused it to break and lose its partial rigidity. In the end, this prototype failed in creating the form that I wanted, but helped progress the state of the development environment software that accompanies the cards, leading to the next phase (prototype 5), where my focus was fully dedicated to the form and fabrication of the cards.

6.5.3 Prototype Refinement

The were countless form refinement done on this version of the prototype. First, I tried using cardboard as the base surface for the components, but the material was not conducive to wiring and it did not have the rigidity/flexibility balance I was looking for in a material, so it could be bent and bounce back upon release. After some searches, I've come across the Adafruit flexible proto-board (a perfboard[8] printed on a flexible substrate) which had the appropriate rigidity/flexibility balance and was suitable for through-hole component soldering. To place the display and micro-controller in their position, I cut a hole in the proto-board so they could be placed and soldered in a consistent position. After some use, I learned that the holes cut in the material was making it brittle and the proto-board would break right below the display board after repeated bends.

A second struggle while working on the form involved finding and routing connecting wires across the card. I tried several kinds of wires: initially, I tried wires similar to jumper-wires which are thin (about 28 gauge) strand wire[9] however the encasing was not flexible enough and caused some connections were tugged during bends and did not feel good in the hand; I then tried solid core wire which had the interesting effect of making the card retain its bent shape (like a metal wire sculpture) until it was unbent, I filed this as something that might be useful in the future, but was not appropriate for my current needs; Last I used strand wire that was encased in a soft silicone cover, which felt pliable enough while holding the card. The connections would still get tugged during bending and cause disconnections, so I added some additional length to every wire so it would be looser. The card on the right in Figure 6.12 shows the excess wires gathered between the display and bend sensors. This solution was somewhat more robust to the disconnections, but involved a mess of wires.

For my third struggle in the form refinements I tried to find successful solutions to the encasing problem. Initially, I poured silicone directly over the electrical components, producing a device with a consistent sturdy-but-pliable texture. I've designed several molds that I 3D printed to help with the pouring process. However, whenever a card device manifested any technical problem, it was impossible to test the components or fix the card without completely disassembling it (and sometime the components would be unusable afterwards). Next iterations were designed to fit on top of the card device like a sleeve. I designed special 3D placeholders that would create the hollow card-sized space while I pour the silicone. When the silicone hardened, I removed the placeholders and inserted the card. However, the sleeve cover would pucker up when the device was bent forward. The texture of the card with the sleeve was not uniform as some areas

---

[8] A perfboard is a board containing pre-drilled holes in a grid pattern, each ringed with a copper pad, they are meant for prototypes that require soldered components.

[9] Strand wire includes many strands of thin wires bundled together in a non-conductive material, as opposed to solid-core wire, which has a single wire wrapped in non-conductive material.

had oddly shaped components. I tried to overcome that last issue by 3D printing rigid attachments for some areas of the card (to be placed under the silicone sleeve) as seen in Figure 6.12 in the center, but ultimately, I decided this design was a failure and came up with a new approach. I document with some detail this part of the process, since I value showing struggles and challenges during the design process rather than painting a picture of perfect progression from one design element to the next.

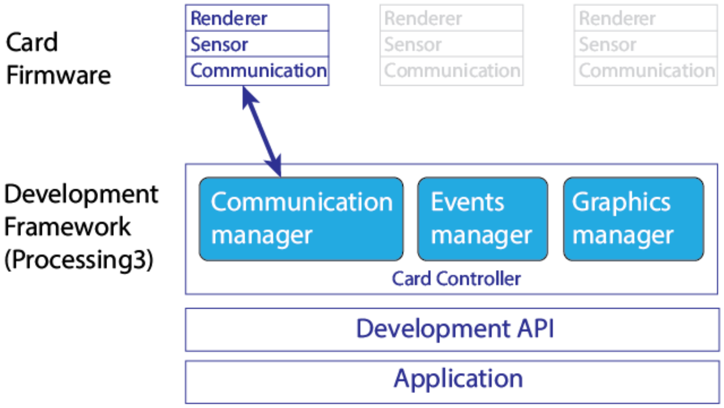


Figure 6.13: Development environment architecture

The work I did on refining the function for this version focused on structuring the development environment for the cards and supporting graphics. The resulting architecture is shown if Figure 6.13. The firmware on each card remains simple: the value of the sensors is read and sent over the communication lines (in this case serial port) to the controller, if there are any incoming messages, the communication function parses them and fires a local function that renders to the display. The main loop of the firmware is as follows:

```
void loop() {
  int val0 = analogRead(FLEX_PIN0);   // read sensors
  int val1 = analogRead(FLEX_PIN1);
  Serial.println("<" + String(val0 - val1) + ">"); // send sensor value

  recvWithStartEndMarkers();  // read messages wrapped with "<" ">"
  if (newData == true) {
    strcpy(tempChars, receivedChars);
```

```
    processMessages();   // render incoming messages
    newData = false;
  }
  delay(20);   // pause and redo loop
}
```

Where the *processMessages* function matches a message code to a function such as *drawCircle*:

```
void drawCircle(int16_t x, int16_t y, int16_t radius, uint16_t color, boolean filled) {
  if (filled) {
    tft.fillCircle(x, y, radius, color);
  } else {
    tft.drawCircle(x, y, radius, color);
  }
}
```

On the development framework side, there are several files that compose the API for the cards, the main classes are the Card, CommManager, EventManager, and GraphicsManager (and some helper classes). A Card object contains its own set of managers: the CommManager holds the serial port connection to the card and is in charge of read/write actions into that channel, the EventManager holds a datastore of several of the last sensor readings that are used to smooth the signal and process it using the state machine shown in Figure 6.10, and the GraphicsManager holds a list of graphic objects to be drawn on the card (so that the manager can send redraw messages of all the shapes if the display was momentarily cleared). Each graphic object can form the message needed for its own rendering. For example, the following is the circle graphic object:

```
class CCircle extends CGraphicsObject {
  int cx, cy, r;
  color c;
  CCircle(int cx, int cy, int r, color c){
    this.cx = cx;
```

```
    this.cy = cy;
    this.r = r;
    this.c = c;
  }
  String createDrawMessage() {
    return "<C,"+str(cx)+","+str(cy)+", "+str(r)+","+red(c)+","+green(c)+","+blue(c)+">";
  }
}
```

And to draw all of the objects in the frame:

```
  void drawFrame(){
    for (int i=0; i<objects.size(); i++) {
      parent.sendMessage(objects.get(i).createDrawMessage());
    }
  }
```

Within the GraphicManager, *parent* refers to the card controller, and *sendMessage* will propagate the argument string to the CommManager. This function would be called to fully redraw the frame after a display clear and may not always be desired due to the slow refresh rate of the display. In addition to the shape graphic objects, there is also an image object which used the SD memory card on-board the Adafruit designed display board. The images need to be loaded before they are called to draw, and the names used must match. The only image format supported by the SD Adafruit reader is BMP24.

Like with any other software development project, the development of the code was incremental and iterative. The documentation in this section is not fully detailed, but should provide a developer with programming background to get a sense of how the code is organized.

6.5.4 Lessons Learned from Prototype

This prototype version was successful on the development framework part, which was deemed sufficient for the minimal viable product I was aiming for, but fell short on the form side, and the fabrication process itself was long, irregular, and produced unsatisfactory results. I therefore decided to change my approach to the card fabrication. I still wanted to use off-the-shelf components and easy to access materials and methods of fabrication, so that the cards can be replicated with little cost and expertise. I have found that some hobbyist use a craft cutting machine on copper sheets to create self-designed circuits, and decided to see if this can be applied to my project.

6.6 Prototype 5 - Slick Design

6.6.1 Prototype Description

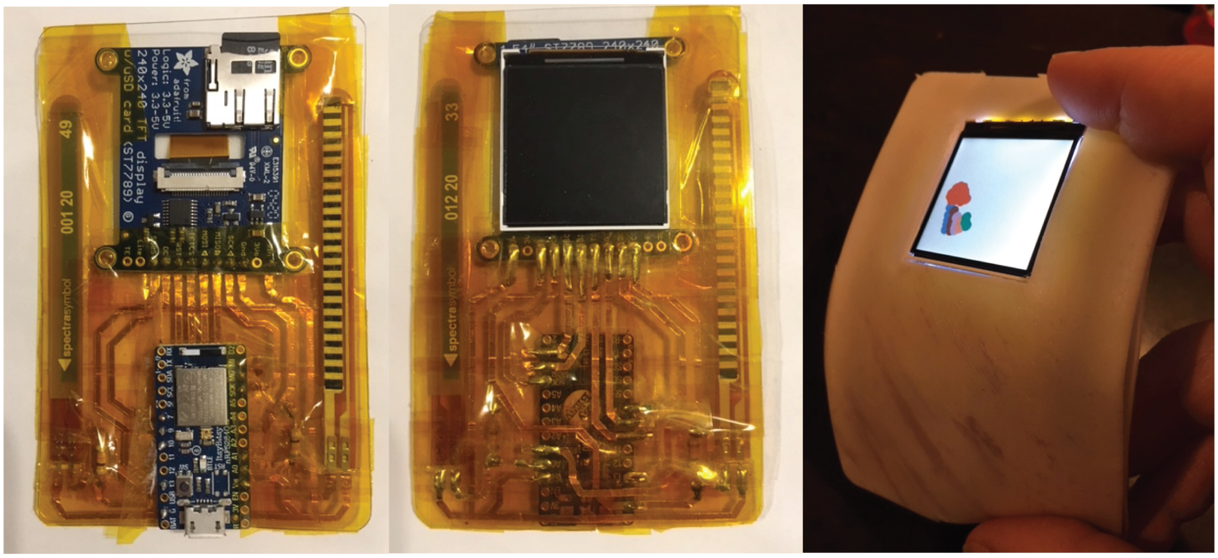


Figure 6.14: Prototype 5 - Slick Design

This prototype version uses similar software to prototype 4, and focuses completely on improving the form factor of the cards as well as improving their fabrication process. In this device, the base surface is an acetate sheet covered with Kapton tape[10] and cut to a specific template using a craft cutting machine (in this case Cameo Silhouette 4). The circuit is cut using the craft machine in two layers - a main layer, and a smaller layer of intersecting lines, that need to be adhered on top of the base using the guides on the card. A small acetate piece separates the two layers. The display, micro-controller, sensors, and resistors all

---

[10] Kapton tape provides electrical insulation.

have bespoke locations, the holes in the base should perfectly align with the components. The components are then soldered onto the base. Instead of a silicone encasement, the card has a cover 3D printed using a flexible filament (SainSmart TPU, 95A Shore hardness), thus guaranteeing consistency in fabrication while lowering the technical experience needed to create a card cover.

6.6.2 Prototype Goal

The goal of this prototype is the establish a finished product that is aesthetically pleasing as well as easy to manufacture by hobbyists. By delegating the detail-oriented work to machinery (the craft cutter and 3D printer) I think that this version successfully achieved this goal with a slicker and thinner card design.

6.6.3 Prototype Refinement

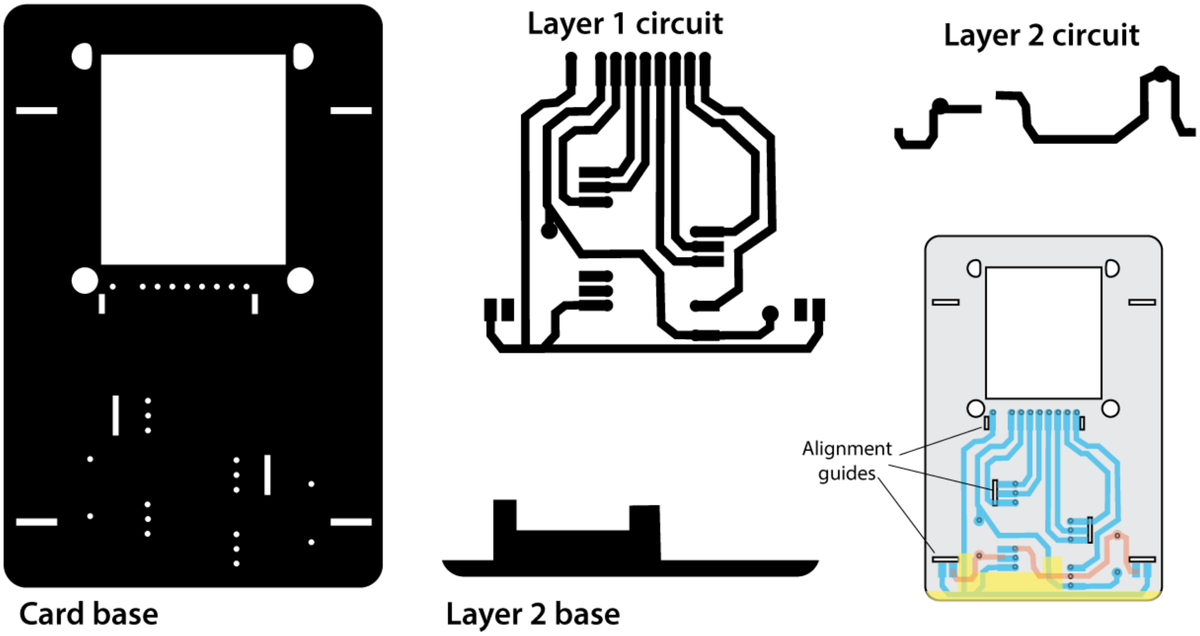


Figure 6.15: Shape files for craft cutting machine

Despite appearing to show a simple and clean final result - the refinement work was not straightforward. I have tried many different variations trying to layout the connections on the card. First versions included a hole for the micro-controller on the base and placed all the wiring around that hole. Placing the micro-controller in this way was supposed to decrease the device's profile thickness, however, it cut the space available for routing wires to a narrow band between the micro-controller and display. Over time, I have decided that the space for the wires is too valuable and the micro-controller was relegated to be fully below the base.

In the beginning of this refinement phase, to avoid crossing wires, I had estimated that I would need no less than 7 layers of circuitry (and insulation between them). This idea was hard to execute, it is complicated enough to align layers correctly - I have never managed to do so seven times. After the decision to eliminate the opening in the base surface for the micro-controller, there was more space to route wires, and by using this space in a new way, I cut the number of layers needed to avoid crossing wires to 2. This was followed by various iterations on what should be on each layer and how to route the wires themselves so as to minimize possibility of accidentally shorting the circuit. Figure 6.15 shows the final design of the base cutout and the two layers of circuitry. The card base has several perforations meant to assist with aligning the circuit layers with the base layer. An illustration on the bottom-right of Figure 6.15 shows as an example how some of those perforations are expected to be flush with the copper traces.

In addition, with new machines and new materials, there is always a need to experiment. The copper sheets in particular are a very delicate material and need to be cut at the slowest speed to make sure that the copper is not peeled off as it is getting cut. The thickness of the wires I could reliably cut was another cause for experimentation. The card in its assembled form is shown in Figure 6.14.

After my work on version 4, and the various problem I had with the silicone covers, both the directly poured ones and the "sleeve" ones, I decided to abandon the silicone material in favor of using flexible filament in a 3D printer. Working with a 3D printer promises a consistent result, which I could not achieve with my manual process of mixing and pouring silicone. It also made it easier to iterate on designs - the printing process might take a few hours, but I did not need to supervise the process, while the silicone process involved multiple stages (and I still needed to 3D print the molds). The general approach to the design of the cover was to make a front piece and back piece that can 1) interlock around the card to secure the electronics, 2) separate to allow removing the card, and 3) remain flexible so as to not interfere with bending. My colleague, Eva Morales Perez, helped me with designs for possible cover.

Figure 6.16 shows two examples of cover designs that follow our design requirements. Both show a 5mm profile. Each design had its advantages and disadvantages. The design at the top of Figure 6.16 has

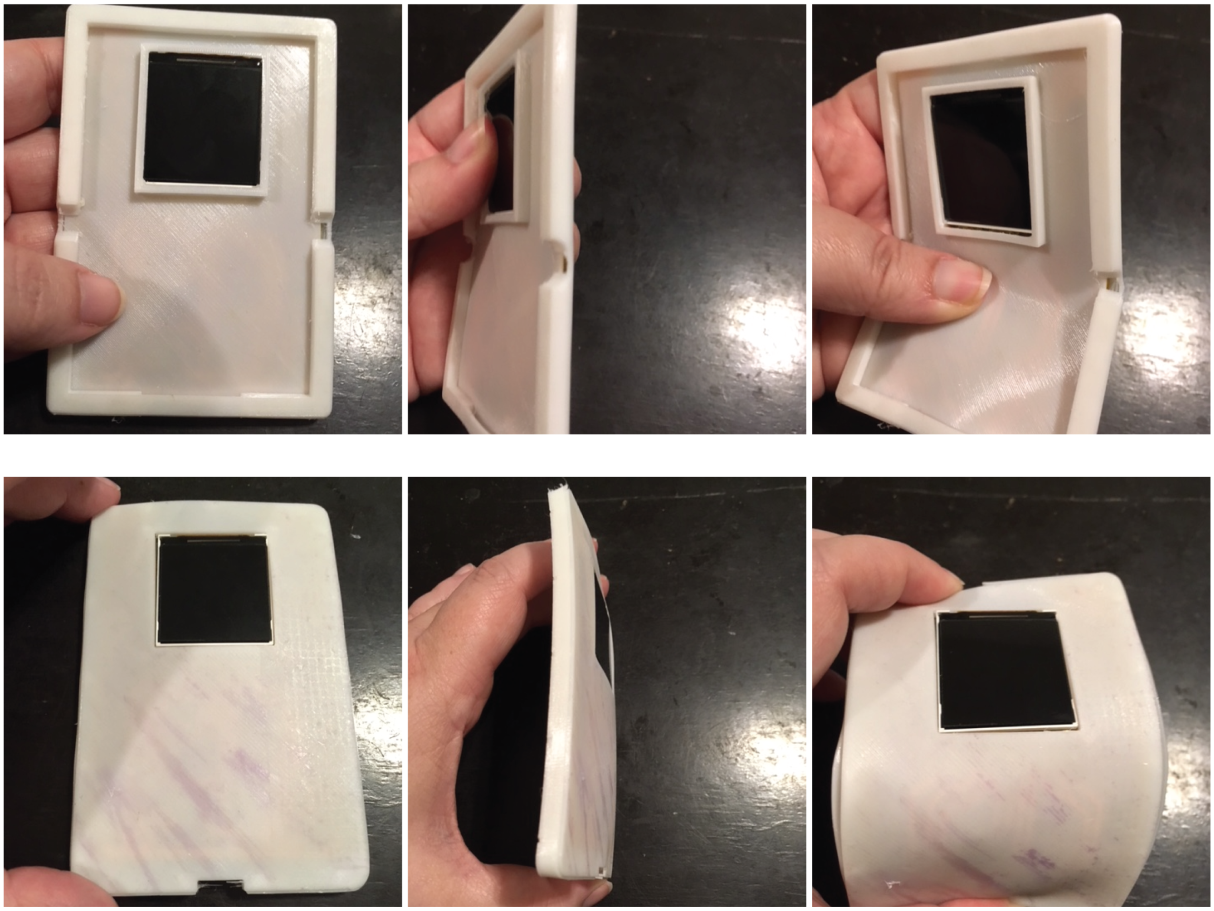


Figure 6.16: Experimentation with cover design

a small gap between the top and bottom parts of the card, which supports smooth bending, however the back piece is bumpy due to the electronics pressing against it. The design at the bottom has a sturdier back piece and enough space between the front and the back pieces to contain all the electronics, providing a smoother feel on the back of the card, however, bending the cover causes bunching up of the material. A clear benefit of using 3D printing for fabricating the cover was how quickly one can try new solution that are vastly different.

6.6.4 Lessons Learned from Prototype

The work on this prototype taught me many new skills which I think will improve my abilities as a designer. The success of this design indicates that it was correct to abandon the approaches I used prior to this version, and think about the problem with a fresh perspective. This design was inspired by various guides and tutorials I found online made by hobbyists creating and sharing their designs with the community.

6.7 Discussion

The last few sections go into some detail over my process of designing the bendable card device and software throughout the PEPA project. Since this is my dissertation, and a chance to demonstrate how I use the DRRD tool to elaborate on all the different aspects of the process, I've gone into a rather lengthy description of my work. A more fitting way to summarize my work throughout the design process is presented in table 6.1.

I created the table based on the lengthier details earlier in the chapter. It shows the full process - through all 5 versions of the prototype - in a succinct way that provides a glimpse of the intentions, struggles, and resolutions that followed every step in the process.

For example, it articulates significant design points like the decision to focus exclusively on bend gestures, decision to generalize the software approach, and the decision to refocus on improving form and fabrication at the later versions. These finer points can be lost without delving into the full process. They represent paths in the design road the are unique to me; a different designer would have gone after different paths.

This method of documentation emphasizes the progression between and within versions of prototype produced. I think of it as a prototype-centric approach as opposed methods that focus on timelines, design-events, specific design ideas, or part-wise development (highlighting the development of each aspect of the design separately, i.e. software design over time, material design over time, interaction over time etc.). There are, of course, advantages to all of these methods, and different situations would be better explained by one over another. This method best explains my process and how I was thinking about it. Future work can show whether this concept is appropriate for the work of other designers. It may also help compare and contrast how different designers work.

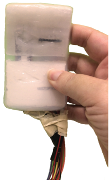
Ver.	Representation	Prototype Description	Prototype Goal	Prototype Refinement	Lessons Learned
1		Simple form: prototype environment with temporary circuitry, components roughly held together in card shape. Advanced function: functioning bend sensors, displays, and network communication, can run a simple game "Spider catch".	• Proof of concept (create digital "cards" that are palm sized, thin, durable, and flexible for playing a game) • Learn how to work with the components • Fulfill requirements for a project-based class	• Focus on building the code for the "Spider Catch" game. • Resolving synchronization and state management between cards	• Complexity and use of memory for a single game was too high - move to passive card firmware controlled by a central controller • Simplify prototype to focus on interaction by temporarily removing wireless communication • Study triggering state actions at different phases of the bend gesture
2		Improved form: prototype environment with temporary circuitry, components encased in a silicone material shaped like a card. Simple function: only functioning components are bend sensors.	• A functioning bendable device for a user study • Measure and record bend signals • Mimic the "feel" of the ultimate card-device	• Implementing an event engine based on theory extracted from the study, recognizing events for start, end, change in threshold, peak bend, and peak release	• A display in required to evoke the card metaphor • The silicone encasment was easy for users to bend • the implemented event model was flexible enough to support several different applications
3		Improved form: some temporary circuitry, micro-controller, sensor, and display components encased in a silicone material shaped like a card. Improved function: event detection software works properly.	• Incorporate all components - micro-controller, display, and sensors - into a single card device with a small form factor		• Soldering is necessary to minimize form factor • Must mount components on a single flexible surface • Micro-controller's cable connector must be placed at the bottom-center of the card • The development code needs to be extended to support graphics management
4		Advanced form: soldered circuitry (using wires), micro-controller, sensor, and display components encased in a silicone material shaped like a card. Advanced function: framework of development environment working including event detection, communication with card, and graphics instructions.	• Ultimate card form: all components on-card in a compact form factor	• Experimentation with – Surface material and form – Wire type – Wire routing – silicone encasing (pour over vs. a silicone sleeve) • Rework development environment to include card object with communication, event, and graphics managers	• The fabrication process for the cards is on the complicated side and lacks consistency
5		Advanced form: soldered circuitry (machine cut copper sheet), micro-controller, sensor, and display components embedded in machine cut card, wrapped with 3D printed flexible cover. Advanced function: framework of development environment working including event detection, communication with card, and graphics instructions.	• Ultimate card form: all components on-card in a compact form factor • Simplify fabrication process	• Experimentation with – Materials and setting for cutting machine – Circuit layout for cutting machine – Flexible materials for 3D printers – Cover designs	• New skills

Table 6.1: Summary of design process

CHAPTER 7
DESIGN WORKSHOP STUDY

As the prototype design reaches a stable point, it is appropriate to explore meaningful way to evaluate it. One recurring methodology within Research through Design is the use of expert interview and their design feedback. With this idea in mind I set out to conduct a study with several of my colleagues who have expertise in game development. This took the form of a design workshop where participants were asked to come up with game pitch ideas for interactive cards. This chapter details the format of that workshop and how I analyzed the data I collected.

While there are many other possible research questions that can frame such an evaluation, I decided to look at the study through the lens of **expressiveness**. For me, this represent a necessary direction for telling the story proposed by the title of this dissertation "research through Design of Bendable Interactive Playing Cards." I ask whether bending interaction suitable, feasible, and expressive enough for interactive playing cards, but in simpler terms, since I thought of the concept "bendable interactive playing cards," I kept wondering: should I make it, can I make it, and if I were to make it - would it be something people can design for? So, turning to look within the design space of *games* for bendable interactive cards, I question the expressiveness of that space. Expressiveness can be stated as a combination of 4 factors: *semantic width*, *semantic variety*, *syntactic width*, and *syntactic variety* (see for example [50]).

For the PEPA game design space, the different gaming experiences that can be created represent the semantic factor of expressiveness, where 'width' stands for the number of experiences, and 'variety' stands for the richness of the difference between them. The syntactic factor of expressiveness would be the basic elements of game that PEPA supports - the cards, the rules, the gestures, the computer actions that can be used. With 'width' and 'variety' standing again for the number of elements and the richness of the difference between them. The 'semantic' experiences are a composition of 'syntax' elements stringed together to form a game.

Of course, since neither the concept of interactive cards nor the use of bendables for designing games have prior theory to support this evaluation, I chose to go with a qualitative approach that will allow me to collect the data first and extract themes from it through the lens of expressiveness. Specifically I chose a

Qualitative Content Analysis (QCA) method. [149, 76]

7.1 Method

7.1.1 Procedure

The workshop was conducted online using the Zoom conferencing tool and lasted just over 2.5 hours. With the exclusion of myself who presented the topics and observed participants (occasionally answering questions), there where 7 participants, though one of them could only arrive at the middle of the second group activity. They were divided into two groups with one group being led by my Graduate student colleague, Dylan Kobayashi, and the other group led by my Graduate student colleague, Kari Noe. Kobayashi and Noe were also group moderators in the workshop (in their respective groups) as well as participants in a pilot version of this workshop. There were multiple parts to the workshop illustrated in Figure 7.1.

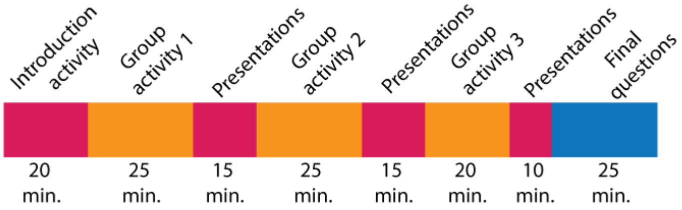


Figure 7.1: Timeline of the design workshop

Introduction

This part included the concept of *interactive cards* and a short activity of reflecting on pros and cons of paper card games as well as pros and cons of digital card games. At the end of this part, group activity 1 was introduced.

Group activity 1

In this part, the two groups worked separately. They were tasked with brainstorming pitch ideas for games for *interactive cards*. They were given a brainstorming guide for designing games (more details below) and were asked to come up with at least one pitch per participant. The prompt for this activity (after the brainstorming guide listed below) was:

> The moderators will write down the results of the randomly generated cues, and a pitch paragraph compiled by the team to describe the game. The pitch should include:
>
> - 1-3 sentences describing the concept
> - What do the cards represent (can be different things for different cards)
> - How would the player/s win
> - In what way the computational component is integrated
>
> The goal is to generate as many ideas as we can.

First presentation

With the two groups joined again, each participant presented the pitch idea that they came up with. This was followed by an explanation about **bend interactions** and how they are implemented the cards I designed. Group activity 2 was introduced.

Group activity 2

In this part, each group was asked to choose one of the pitches they came up with in group activity 1and refine it into a more complete design idea that is intended for *bendable interactive cards* using the bend interactions they were shown. The prompt for this activity was:

> Choose one idea from part 1 and fully develop it. Discuss which idea you think will work best with bend gestures. Specify:
>
> - The rules of the game (this would translate into its logic)
> - Which gestures are used
> - How would the game inform the players of possible gestures
> - What would be the feedback for actions, during and after gestures are made.

- The resources needed for the game, such as art or sound assets.

Your moderator will write the details as you develop them.

Second presentation

With the two groups joined again, each group presented their detailed bendable card game idea. Group activity 3 was introduced.

Group activity 3

In this part, the two groups worked separately. They were tasked with brainstorming pitch ideas for games for ***bendable*** *interactive cards*. This was similar to their task in activity 1, except that they now had to keep the gestural interaction in mind. They were given the same brainstorming guide for designing games and were asked to come up with at least one pitch per participant. The prompt for this activity was the same as above (in activity 1).

Third presentation

With the two groups joined again, each participant presented the pitch idea that they came up with.

Final questions

In the last part, I addressed questions about the experience of the workshop to the group and prompted them for feedback. The questions (paraphrased) are:

- How do you compare your activities in the first (not bend gestures) and third (with bend gestures) group activities?
- Did you find the ideas of other participants in the workshop stimulating?
- Which part of the workshop was to hardest?
- Are interactive playing cards a good idea?
- Would interactive card games benefit from having a different form of interaction (not bending)?

- Do you feel that bending interaction could benefit some interactive card games?
- While designing, how did you perceive the cards in your imagination (paper, digital)?

7.1.2 Brainstorming Guide

The structured activities were based on boardgame brainstorming randomization tables from Cudo website[1]. This site offer random tables for themes, mechanics, and win condition. Participants were instructed to use online dice to generate random parameters to guide their brainstorming activity. The random parameters include:

- **Number of players** (roll a 1-6 die, odd number means 1 player, even number means 2 players)
- **Number of cards** (roll a 3-15 die, the number rolled is the number of cards to be used)
- **Game themes** (roll a 1-100 die twice to get two themes from the game themes table 7.1)
- **Game mechanics** (roll 1-20 die to determine a mechanic from the game mechanic table 7.2)
- **Win condition** (roll 1-8 die to determine how the game ends from the win condition table 7.3)
- **Computational component** (roll 1-8 to choose an aspect of the game that may necessitate the use of digital cards from the computational component table 7.4)

Following the pilot version of the workshop, it was decided to be more flexible with the themes of the game, asking participants to roll 4 times and choose their preferred two themes out of the four options.

7.1.3 Introducing Bendables

Before the second activity, the participants in the workshop were briefed about bendable interaction. First they were introduced with the following text:

> For a while, there has been some research into the idea of using bending as a form of input. Bending is natural (think of bending pages in a book). The bend gesture has several characteristics that make it interesting:

---

[1] `https://www.cudoplays.com/blog/board-game-brainstorm-the-cure-for-game-designers-block` (viewed 10/4/2021)

1	Steampunk	26	Anime	51	Shopping	76	Family
2	Office	27	Woodland & Creatures	52	Duel	77	Beauty
3	Math	28	Fantasy	53	Treasure	78	Natural Disasters
4	Zombies	29	Horror	54	Monsters	79	Sasquatch
5	Space	30	Mystery	55	Emotions	80	Music
6	Industry	31	Art	56	Loud noises	81	Action Movies
7	Cooking	32	Cars	57	Big city	82	Garbage
8	Fashion	33	Wizards	58	Language	83	Merchants
9	Babies	34	Caves	59	Ducks	84	Politics
10	Secret Agent	35	Kittens	60	Sports	85	Insects
11	Farming	36	Spiders	61	High School	86	Crime
12	Construction	37	Computers	62	College	87	Bathroom
13	Inanimate Objects	38	Swords	63	Costumes	88	Hillbilly
14	Sea Creatures	39	Plants	64	Dancing	89	Friendship
15	Ocean	40	The Olympics	65	Hair	90	Dragons
16	Robots	41	Volcano	66	Clothes	91	Rebellion
17	Dinosaurs	42	Celebrities	67	Jewelry	92	History
18	Construction	43	News	68	Guns	93	Aliens
19	The 1920s	44	Gangs	69	Comics	94	Unicorns
20	Books	45	Drugs	70	Science	95	Murder
21	Retro Videogames	46	Religion	71	Superheroes	96	The Moon
22	Smart phones	47	Roller blades	72	Hawaii	97	Fairy Tale
23	Romance	48	Holidays	73	Party	98	Television
24	Pirates	49	Flying	74	THE INTERNET	99	War
25	Ninjas	50	Science Fiction	75	Millennials	100	Furniture

Table 7.1: Game Themes

1	Area control / influence	11	Route building / network building
2	Auction / bidding	12	Speed / real time
3	Card drafting / hand management / set collection	13	Dexterity
4	Deckbuilding / whatever-building	14	Action programming
5	Dice rolling	15	Performance (singing/acting/charades etc.)
6	Movement	16	Team-based or asymmetric
7	Memory / pattern recognition	17	Bluffing / hidden role
8	Tile placement	18	Trading / negotiation
9	Worker placement	19	Betting / wagering
10	Press your luck / risk management	20	Simultaneous turns / actions

Table 7.2: Game Mechanics

1	Most points / resources
2	Last person standing
3	First to the finish line
4	Use up all assets
5	Popular vote
6	Solve the mystery
7	Collect a set / build a thing
8	Complete quests / assignments

Table 7.3: Win Condition

1	Audio
2	Randomization
3	Timer
4	Animation
5	Conditional rules
6	Secrets
7	Procedural generation
8	AI

Table 7.4: Computational Component

- location on device - devices may be able to detect bends on the corners of the device or at it center, horizontally or vertically
- direction (up/down) - in bends, one direction is in opposition to the other direction of bending, making it appropriate for actions and their negations
- size of bend area - a flexible surface can be bent so that only a small area is "lifted" from the plane, or so that a large area is "lifted" from the plane
- angle of bend area - inputs read through bend sensors are continuous in nature, their values can be transformed into an angle that measures the relative rotation to the original rotation (around the axis of the bend) speed of bend - we can measure the quickness of the gesture, it is independent of the angle of the bend, a device can be bent between two specific angles quickly or slowly
- duration - this can be the length of time a bend gesture is maintained at a certain angle, or the overall time for the whole gesture

Participants were also **1)** shown a video of the PEPA cards in interaction, **2)** had a reference sequence of still images depicting a simple bend cycle (Figure 7.2), **3)** had an image showing how multiple cards can be used for interaction (Figure 7.3), **4)** had a reference table listing possible bend events, their description, and what kind of additional data they can expect the system to know for each event (i.e. bend direction, level of bend, duration since start of gesture etc.).

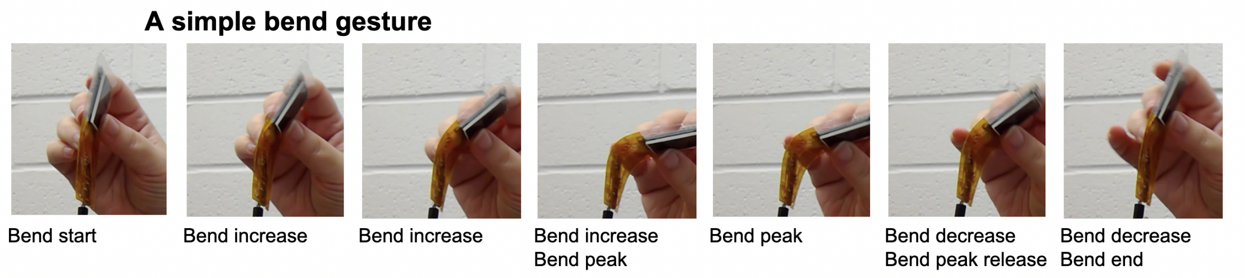


Figure 7.2: Images of a card being bent

7.1.4 Data Collection

The workshop was video recorded by all team leaders (with the permission of participants). In addition, team leaders and participants had access to Google documents allocated for the different group activities in which to write their thoughts and a google Jamboard to sketch as needed. I transcribed all of the recorded material to support the generated documents in the analysis phase.

7.1.5 Data Analysis and Coding Frame

The QCA process I applied to the data involved the following steps:

1. Read through the material to get familiar with it
2. Identify analysis units and coding units
3. Identify themes in the coding units
4. Form a coding frame with dimensions and categories identified in previous step
5. Correct and iterate the coding frame
6. Apply coding frame to the data

The three core dimensions I identified through the iterative process of coding are: *Computer role*, *Gesture usage*, and *Use of prior knowledge*. Table 7.5 details the coding frame that was developed from the data including the categories identified for each dimension, a description of each category, and some examples.

Bend one card while holding several

Stack several cards and bend them together

Select two specific cards and bend them, can use any of the bend events for each card separately

Figure 7.3: Variation of bending with more than one card

For each category, acceptable values were ’present’ or ’not present’ that indicate whether the category was represented in some way during the pitch process (including group discussion, written text, and presentation to the general group), or not. The data was coded individually by myself and by Kari Noe following the coding frame as a guid. After our individual coding we regrouped and discussed all issues on which we disagreed until we reached an agreement on all points.

In the first dimension, computer role, the categories identified are manage "busy" work, simulate behavior, and perform "magic". These categories were inspired by Rogerson’s et al. "Digitising Boardgames: Issues and tensions" [138]. In this paper, the authors discuss using the metaphor of ’Boardgame’ in digitized versions of boardgames and they bring up the enhancements afforded by digital versions, such as, sound and animation, that are in tension with the boardgame metaphor. These enhancements are "magical" functions that are not available in traditional versions of boardgames, hence the preform "magic" category. Rogertson et al. [138] also discuss at length *Articulation work*, an expression borrowed from collaborative work, that refers to the chores, or housework of the game, *"the work necessary to make the play happen"*. The chores of the game can include game setup, keeping scores, upholding the rules, and managing turns, these can be easily implemented digitally, however, digitizing them can take away from social aspects of the game as some players consider chore to be fun [180]. Nonetheless, taking away chores from the player may allow them to focus more on the game and effortlessly follow the rules. I categorize such activities as "busy" work. Simulated behavior does not exist in the boardgame metaphor, so it could be viewed as an instance of "magic," however, as I noticed simulated behavior manifested in several game pitches without being required, I decided it was a sufficiently interesting category to separate from other forms of "magic".

Gesture usage was a necessary dimension since the workshop's focus is on ideating games with and without gestural limitations. Due to the shortness of the workshop, ideas for game remained mostly underdeveloped, and did not specify interaction throughout the game (even when asked to provide it by instructions). For that reason, the categories in this dimension were left fairly abstract. The central point of consideration was Buxton's concept of *phrasing* [22, 23] which compounds several micro-tasks into one phrase gesture. For this coding frame, I allowed a "single gesture" category to capture the most simplistic mentions of gestures (i.e. when participants say "tap" or "bend" without further elaboration), and a phrasing category to encompass any other interactions that are described as game actions without specific details as well as clear phrases such as "bend and hold". The exception to the single and phrase gestures emerged when reviewing the data and identifying gestures that are a sequence of gestures that are not known in advance, such as, increasing/decreasing bend level based on dynamically changing visual cues on the screen to preform an action. These can still be viewed as a phrase since they represent a single action, however, they require cognitive processing and adjustment of movement while they are preformed (vs. complex gestures that can be easily executed with muscle memory) so I separated them to a different category I call dynamic gesture.

While going through the material, I observed that participants would frequently bring up names of games to explain what they mean in their pitches, this is actually very common in game design - to rely on knowledge from existing games or other activities while ideating for game design [61]. This is the source of the prior knowledge dimension. Some participants used their experience from prior knowledge to explain the game concept or the game interaction, which were marked in the corresponding categories. A third category "cardiness" was inspired by the boardgame metaphor touted by Rogerson et al., but narrowed down to refer only to card games and the affordances they provide in the real world, such as, using and shuffling a deck of cards, spreading cards on a surface, holding a "hand" of cards, as well as the use of cards to represent elements with attributes or abilities/effects. The underlining question for this category was "Can you imagine these as actual cards?" Some pitches did not indicate this category, showing the cards were viewed more as devices than cards for those ideas.

7.2 Results

7.2.1 Generated Game Pitches

This section will present several example from the pitches generated at different workshop activities. Table 7.6 shows the properties randomized for all the pitches as well as the participant who created them. Participants were asked to follow these properties as starting-off points for their own ideas. (Items marked * were not possible according the instructions and resulted from misunderstanding or confusion.)

Turf Wars

This pitch was created by D2 during activity 1.

> In Turf Wars, the players play two competing gangs fighting for turf. They need to work collaboratively together to stop an encroaching 3rd gang, played by *"the game itself"* from getting territory, while still increasing their own turf, since the player with the largest area under control will be the winner after they get rid of the 3rd gang. The cards represent resources and turfs that the players can trade with each other. However, in line with the "secrets" computational component, some resources and turfs are traps that can backfire on the player.

Murder and Zombies

This pitch was generated by K1 during activity 3.

> In this game, the player has 6 cards positioned in front of them in such a way that there are multiple paths to the player. A horde of zombies is lurching toward the player. The player

Category	Description	Examples
Computer Role		
manage "busy" work	actions that are administrative of the game and not part of the play actions such as, managing turns, game setup, enforcing rules, keeping game pieces in order, rolling dice, randomization, etc.	*"the traps that are on that card are randomized every time you use it's a new one"* *"at the end of the game the player with the most reputation gets to live with the aliens"* *"it could essentially roll an internal die, to calculate that?*
simulate behavior	represent intelligence that has effect on the gameplay (ai character, npc)	*"there is a third player in the sense [that] there is the game"* *"I know I have one player but then it had like first to finish-line so this player might be an AI"*
perform "magic"	things that could not be replicated by a person doing busy work, or by people taking the task of simulated people, includes rich media and effects, secrets/hidden information ("magical" functions - offer enhanced functionality not available in the literal version of the gameboard metaphor)	*"maybe whatever sound effect is louder and it gets softer when you are in the sweet zone because then you're safe"* *"some resources and turfs are traps, so that's the secret part"*
Gesture Usage		
single gesture	A single gesture that has impact on the game (i.e. bend to select)	*"the player can bend each individual card to have the screen playing to 'bend the truth' of the newscaster"* *"you can just bend it to cycle through your items"*
phrasing	a sequence of gestures that is known to the player before they start performing it (a bend-hold is an example of a phrase, but they can be more complex)	*"to use these resources you have to pick it up and apply it to the blueprint"* *"a long bend to discard an item in your inventory"*
dynamic gesture	Often is a kind of minigame, when a sequence of gestures is expected, but the player does not know the expected sequence when starting the gesture	*"you have to bend in the correct time to make a goal"* *"you're trying to bend the card a certain way to keep the slider in the sweet zone, but the slider is continuously trying to move out of it"*
Use of Prior Knowledge		
explain game concept	Mentioning something from "the real world" (other games or anything else) to explain something about the game's concept (explaining the idea behind the game)	*"think of it as a very weird way to play like bop-it where you have to keep on quickly doing things to continue on with the round"* *"little quests involve a game of Simon says"*
explain physical interaction	Mentioning something from "the real world" (relating to other games gameplay, other devices, or anything else) to explain physical interaction (explaining the action behind the game)	*"is something you know in fishing games when suddenly you have to move"* *"so as you walk, bend that card to shake that booty"*
"cardiness"	Mentions properties associated with paper card games (or physical board games) such as a use of a deck/die, organizing cards in space or having a "hand", association of cards with a specific type of element (Can you imagine these as actual cards?)	*"I can choose to change one of my cards as opposed to discard and pick it up, but it's the same function"* *"each card represents a part and each turn you have to choose a card to swap out"* *"8 cards on the field so-to-speak representing resources"*

Table 7.5: Coding frame

Themes/name	Designer	# of players	# of cards	Game Mechanic	Win Condition	Comp. Component
murder cave	D1	2	3	risk management	collect a set, build a thing	AI
murder cave (mod)	D1,D2,D3	2	3	risk management	collect a set, build a thing	AI
space turf	D2	2	9	trading/negotiation	solve the mystery	secrets
workspace woes!	D3	2	10	dexterity	most points/resources	randomization
ocean & high school	K1	1	6	action programming	first to finish line	randomization
treasure & duel	K2	1	6	memory/ pattern recognition	complete the quest	secrets
robots & beauty	K3	2	4	team based asymmetric	last person standing	audio
robots & beauty (mod)	K1,K2,K3,K4	2	4	team based asymmetric	last person standing	audio
fairytale moon band	D1	2	4	auction/bidding	most points/resources	audio
sportsNews	D2	1	1*	memory/ pattern recognition	complete the quest	procedural generation
FakeNews	D2	1	9	memory/ pattern recognition	complete the quest	procedural generation
guised runway	D3	1	8	bluffing/ hidden	popular vote	animation
murder & zombies	K1	1	6	area control/ influence	last person standing	conditional rules
language & sci-fi	K2	2	7	memory/ pattern recognition	most points/resources	conditional rules
guns & forest animals	K3	6*	5	auction/bidding	solve the mystery	AI
alien wizards	K4	1	7	movement	popular vote	animation

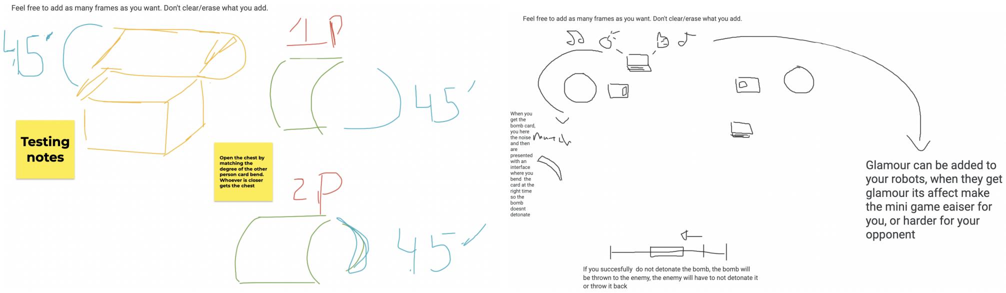
Table 7.6: Pitches and their random properties

> needs to deploy traps to kill the zombies before they reach them. Each of the cards showing the trails (and zombies) will show possible traps that randomly appear that the player can deploy to get rid of zombies in that area. The way the trap is deployed changes based on the kind of trap shown on the card, each type of trap is deployed with a bend sequence associated with that type. After a trap is deployed, a new random one appears in its place. The game ends when the player survives and has killed all the zombies, or the zombies get to the player.

Murder Cave (mod)

This is the modified version of Murder Cave as it was developed during activity 2 by D1, D2, and D3. (Figure 7.4 shows associated sketch.)

> In Murder Cave, two players are stuck in a cave and need to build a contraption that will help (one of them) to get out. They build this contraption with items they find in the cave, however, some items, especially the really useful ones, may kill the player instead of helping them. Each player has 3 cards: the first represents a treasure chest, the second represents the player's inventory, and the third shows the player's point of view in the cave. Upon finding an item, the players need to compete to see who will get it: first, one player



Feel free to add as many frames as you want. Don't clear/erase what you add.

Open the chest by matching the degree of the other person card bend. Whoever is closer gets the chest

Feel free to add as many frames as you want. Don't clear/erase what you add.

When you get the bomb card, you here the noise and then are presented with an interface where you bend the card at the right time so the bomb doesnt detonate

Glamour can be added to your robots, when they get glamour its affect make the mini game eaiser for you, or harder for your opponent

If you succesfully do not detonate the bomb, the bomb will be thrown to the enemy, the enemy will have to not detonate it or throw it back

Figure 7.4: Sketches from Activity 2

> bends their treasure card (without showing the other) and the other player tries to mimic that bend, then the second player bends their treasure card and the first player tries to mimic that bend, whoever came closest to the bend that was secretly created by the other player, gets the item. Next, each player goes through their inventory card, they can bend backwards to cycle through the inventory, and use a long bend to discard an item from the inventory. Following that phase, the player moves to their point-of-view card and the inventory decides if any of the items would try to kill the player. If any items attack the player, the player will see those items fly at them and can flick them away. A successful flick will mean the player dodged the attack, otherwise, the character dies.

This version is much more complex than the unmodified version of the game from activity 1, where each of the 3 cards represented one item and the players need to pick up and swap items until they have a combination of 3 that can help them escape the cave (or they get killed by an item). The modified version lost some of the simplistic cardiness of the original idea as the designers tried to find a variety of different ways to use bending, however, they did come up with an interesting mechanic of performing a hidden gesture that your opponents needs to guess. The designers used the metaphor of a key code to think of it as opening a chest. The very idea of opening a treasure chest emerged from them likening the bend gesture to the top of a chest opening.

Robots and Beauty (mod)

This is the modified version of Robots and Beauty as it was developed during activity 2 by K1, K2, K3, and K4. (Figure 7.4 shows associated sketch.)

> In this game, players are trying to build-up and beautify their robot while avoiding possible bombs that would undo their work. Each player has two cards, one is used to represent the robot and its status, this card will be updated when you pick up "bling" or after an explosion, and the other card represents the "deck". The status of the robot card can be reviewed by bending the card to look through all the collected items and see what they do. At the beginning of a turn, a player bends their deck card to draw - they will draw a bomb or a boon. Boons can be added to your robot and help you against bombs (or worsen your opponent's situation). Bombs need to be diffused. When the card is drawn to reveal a bomb, the player will hear music and the sound of ticking and the card will change to a kind of mini-game showing a slider with a target that moves along with the bending of the card. As the bomb ticks, the player tries to keep the target within a "sweet-spot" on the slider, but the target keeps trying to move out of it. If the player succeeds in keeping the target in the sweet spot when the music stops, the bomb is diffused, otherwise, it explodes and your robot suffers damage. If you successfully diffused the bomb, it will move to the other player when it is their turn to draw. If your opponent diffuses the bomb as well, it will continue to pass between you with increasingly harder difficulty, until it explodes. The boons can make this phase easier, for example, by making the target slower or the bending less sensitive. The last player standing, wins the game.

This modified version stayed closer to its original pitch, with most of the designers' time focused on the actual process of diffusing a bomb and what would happen to it afterwards. The likened the passing of the bomb to a game of "hot potato" and the slider type mini-game is a type of mini-game common in many existing games, soccer and fishing were brought up as examples of similar type of mechanic by one of the designers.

7.2.2 Coded Results

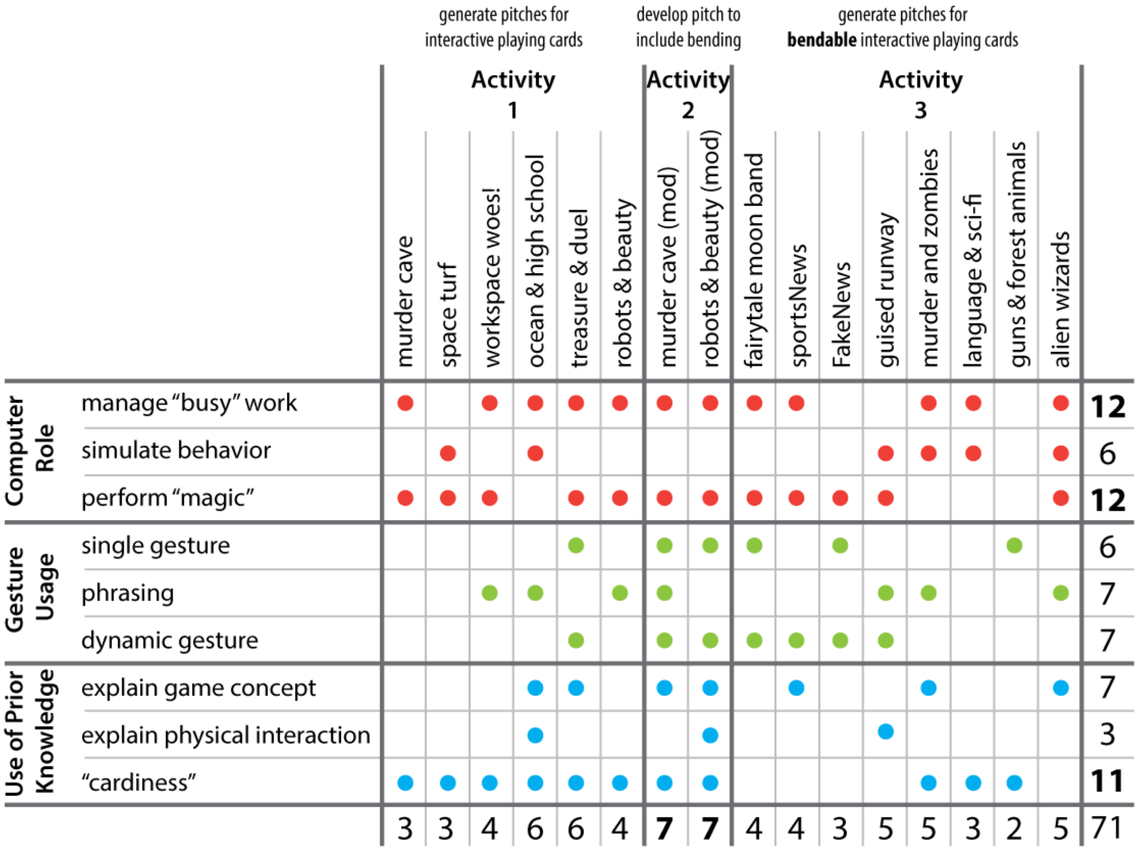
Using the coding frame specified in Table 7.5 we have generated the results chart 7.5 marking the presence of of a category for each pitch with a dot. The pitches are divided by the activity they were generated in. Activity 2 produced an elaborated version of two of the pitches from activity 1, so, perhaps unsurprisingly, they included more descriptions of gestures then their activity 1 sources. We also see that the most frequently present computer roles across all pitches include managing "busy" work and performing "magic", two roles that are inherently connected to digitizing boardgames. Another highly present category is the use of "cardiness" properties, though this property was less prevalent in the pitches generated in activity 3. These pitches that did not evoke "cardiness" were judged during the coding process to be likened to digital toys, those pitches involved complex gestural descriptions which were preformed on cards (in the pitches) but could have easily be adapted to other form such as the Siftables platform [119]. The following section go into further discussion.

7.2.3 Computer Role

The category of "busy" work was very common with 12 out of 16 units of analysis containing some references to randomization or keeping track of game score. The pitches that did not have evidence of this category mostly had very partial descriptions that focused on a different aspect of the game (such as use of AI or secrets), so it is likely that the designers did not have the time to express how busy work would be conducted for these games.

Simulated behavior, a category that included the use of AI or NPC (non-player characters), was exhibited in 6 out of the 16 pitches. It should be noted that none of the pitches that were marked to present this category had the AI as their computational component (and the two pitches that rolled AI, murder cave and guns and forest animals, did not end up showing evidence for this category). Simulated behavior are common, though not necessary, for digital games, but cannot be efficiently implemented in traditional board games.

Perform "magic" was coded present for 12 out of 16 pitches, but would very likely be present in each one as it would include animations and sound which are ubiquitous in digital games. However, the unmarked pitches did not contain details that would indicate how magic would be incorporated.



		generate pitches for interactive playing cards						develop pitch to include bending		generate pitches for **bendable** interactive playing cards								
		Activity 1						**Activity 2**		**Activity 3**								
		murder cave	space turf	workspace woes!	ocean & high school	treasure & duel	robots & beauty	murder cave (mod)	robots & beauty (mod)	fairytale moon band	sportsNews	FakeNews	guised runway	murder and zombies	language & sci-fi	guns & forest animals	alien wizards	
Computer Role	manage "busy" work	●		●	●	●	●	●	●	●	●			●	●		●	**12**
	simulate behavior		●		●								●	●	●		●	6
	perform "magic"	●	●	●		●	●	●	●	●	●	●	●				●	**12**
Gesture Usage	single gesture					●		●	●	●		●				●		6
	phrasing			●	●		●	●					●	●			●	7
	dynamic gesture					●		●	●	●	●	●	●					7
Use of Prior Knowledge	explain game concept				●	●		●	●		●			●			●	7
	explain physical interaction				●				●				●					3
	"cardiness"	●	●	●	●	●	●	●	●					●	●	●		**11**
		3	3	4	6	6	4	**7**	**7**	4	4	3	5	5	3	2	5	71

Figure 7.5: QCA coding on game pitches from the workshop.

7.2.4 Gesture Usage

The gestures were present sporadically across the board. In the first activity, we saw 1 single gesture, 3 phrasings, and 1 dynamic gesture out of the 6 pitches in this phase. It is not surprising to see such sparse details about gestures in this activity, since participants were not asked to focus in interactivity, which allowed them to describe the game from a higher level. The gestures that were present included a "tap" single gesture in a "Simon says" kind of game (which as a sequence was also marked as a dynamic gesture), and abstract phrases like *"you have to pick it up and apply it to the blueprint"*, *"you can put status effects on other characters"*, and *"you can pick 4 cards at a time."*

In the second activity, designers were specifically asked to include bend gestures in the game, so we see more gesture types present for each of the 2 pitches (robots & beauty missing only the phrasing category).

In the third activity, designers remained aware of the gestural component of their game pitches and only 1 out of the 8 pitches in this phase did not present some form of gesture. Most commonly, the dynamic gesture category was present in pitches from this activity as designers were trying to think of meaningful ways to include bending into a game, for example, *"you have 4 cards, each card has a different instrument on it and they're playing the song at a different tempo, so you have to bend the card to try to get the tempo to match, once you get all 4 tempos to match you pass that level."*

7.2.5 Use of Prior Knowledge

From the 3 categories of prior knowledge to explain game concept, physical interaction, and "cardiness," only the last one seems insightful. The prior knowledge to explain the game concept is sporadically used among the pitches, 7 out of the 16. Games like the Pokemon card game and "Simon says" were used to explain the concepts behind games in the first activity. In activity 3, one of the games was compared to "Wario Wars", another used the example of "Bop-it" to explain how the player would need to perform actions quickly, and a surfing competition where surfers get scored for the tricks they preform explained the scoring mechanism in a third game. The two games that were modified in activity 2, didn't originally include references to prior knowledge, but as the groups spent more time with a single idea, comparisons were unavoidable, *"like the caterpillar from 'New Mario Party'"* or *"like Hot Potato?"*

Surprisingly, there were very few cases of using prior knowledge to explain physical interaction, only 3 out of 16.

The "cardiness" category is interesting as all pitches from the first and second activity exhibited some form of cardiness: *"the cards are resources... they have to trade"*, *"each player gets one card for their hand"*, *"think of the cards as a deck."* In activity 3, however, we see some instances of cardiness (3 out of the 8 pitches in this phase), but the other 5 pitches were found to be more toy-like than card-like during the coding process as the descriptions revolved around the manipulation of each devices more than anything related to a card game metaphor.

7.3 Discussion

7.3.1 Impressions on Interactive Cards

In the course of the workshop, the designers came up with a variety of plausible and diverse ideas. Their ideas showed signs of utilizing the computer for multiple roles. Kosa and Spronck [98], in their investigation into positive (and negative) attitudes towards enhanced boardgames, reported that according to their content analysis, players thought such games have potential to decrease tediousness, enhance enjoyment, and provide designers with new tools, such as, number of players, cooperation, secrets or hidden information, multimedia effects, randomization, online updating, as well as AI, NPC, dynamic difficulty adjustment, and procedural generation. The pitches generated in the workshop indeed manage to put many of these new tools into play. On the other side, one of the negative attitudes reported in the paper stems from lack of tactile feeling. The solution for this issue is represented by "cardiness" which bridges the gap between *a device* and *physical cards*.

Unfortunately, the system offers no solution to other negative point discovered by Kosa and Spronck, as they mostly involve a dislike of having to rely on digital devices as part of gameplay.

The biggest caveat for designers in the workshop regarding the interactive cards concept, involves understanding the virtual space represented by the digital property of the device, and how to balance it with a desire to force "cardiness", to make the game use enough 'game bits' [139] to make it feel like a traditional card game, with all the benefits that follow. As one designer said:

> k1: *"There's a weird line of trying to think 'ok, technically this one card can play the whole game because it's a screen and I can keep changing the interface', but also when we think of card games... deck could have 52 cards... so we're also trying to think... why do we need more cards? ... what's the sweet spot between not relying so heavily that this card is going to do everything for us?"*

When designers managed to see the space of virtual cards, they use words like "draw a card", "swap cards", "discard', and "deck". When they couldn't, namely, in the 5 pitches generated in activity 3, that we rated as toy-like - there is nothing beyond the physical appearance of the imagined card that resembles cards; These games could have been created for a toy platform like Siftables [119] by exchanging bending interaction with tilting or shaking, and the spirit of the game would stay the same.

7.3.2 Impressions on Gestures and Bending

When asked how they compare their experience in the first, second, and third activities, designers were unanimous: the first activity was easier as they felt less restricted. In the third activity, many of them felt stumped and unsure about how to incorporate bending interaction. This may have led them to the toy approach mentioned before- the focus on the bend manipulation, overshadowed the objective of making a card game. As they say:

> D1: *"You would think it would be the other way around, but it was easier to just think of the game first, then think about how to add the bending in."*
>
> D3: *"I feel like the intention has changed ... in part 1 the intention was "how do we make a game using these things", whereas part 3 the intention is 'how do we showcase the bending aspect with these things that I'm given'."*

The designers were also concerned about how unfamiliar the gestures are, K1: *"I have a thought that touch would probably be ... easier, not so much in the sense that it is easier, but just that people are more used to touching things."* This led them to consider possible reasons to prefer bending over touch interaction:

> D3: *"Why would I want bending over touch? ... if you are doing a two player game that uses subterfuge ... how do you interact with your cards without being obvious, right?... If you bend it, then you're really obviously bending, and the other player can react to it... But if you're trying to hide it, you can do small slight bends."*
>
> K1: *"A good design value to have is that whatever the movement of the bending it should be satisfying ... there's something about that movement that will make a satisfying experience for that interaction.*
>
> D3: *"...playing into analog tactile ... you have a finer control - physical - because you have that direct one to one manipulation."*

> D1: *"you guys have a good example... you could do it with a slider, but it is kind of nicer to do it with the bend, especially if you have haptic feedback, for when you are in the sweet spot. I feel like that would be more satisfying than just like a slider. "*

Simply put, they suggest that the possible advantages of bending over the more familiar touch interaction include **1)** Finer physical control for continuous control over values, a topic that was explored by prior works in bendables, for example, Bendy [107] and FolfMe [86] use bends for such application **2)** Have a satisfying experience from the gesture itself, for example, FlexSense [135] introduces a sheet overlaying a table that can be pulled up to mimic peeking through pages, **3)** Possibility to add haptic, for example, MimicTile [120] controlled the hardness of the device and used it to indicate a maxed-out zoom level, and **4)** To explicitly show or hide interaction with device, this is a social aspect of the gesture that I believe remain un-investigated, a project exploring password entry for the blind [38, 18] touched on hiding the gesture, but there is no work looking at the gesture as a way to cue interaction to collaborators.

As we concluded our discussion on the bendable interaction vs. touch (or other) interaction, K4 reminded us: *"Not all games are going to be good for bending, not all games will be good for something that's not bending, right?"* Which led to the following conversation, reminding us that just because an input technique is implemented, that doesn't mean that people will use it:

> D2: *"Take the Switch™, it has a touch screen. How many games actually use that touch screen?"*
>
> K4: *"Yea, that's a very good point."*
>
> D1: *"I actually forgot it had a touch screen."*
>
> K4: *"Everybody forgot. Because nobody uses it. Because there are no games that actually make use of it, really."*
>
> D2: *"The only time you end up using it is when you're putting your wi-fi password."*
>
> K4: *"Yea, that's it. For the menus."*
>
> D1: *"Super true."*

7.3.3 Workshop experience

According to post workshop discussion, participants had a hard time with some of the themes that they encountered, to the extent that when they were having a hard time in activity 3 they were not sure if the difficulty was due to the added condition of interaction or if they just had really bad luck rolling properties they were "stuck" about. It was possibly a combination of both, however, it was suggested that having to choose just one theme might have worked better.

> D1: *"I feel like the three of us just had way worse topics second time around which is just random."*
>
> K1: *"I think the hardest part, at least for me... it's just that we had to match these things like themes ... I think that made me stuck on that a while longer than it needed to."*
>
> D1: *"Sometimes you get 4 good themes, and sometimes you get 4 awful themes... But it... it is nice to give you a quick direction to go."*
>
> K1: *"Maybe just one theme is better than two themes because you tend to just stick to one stronger one anyway."*

Similarly, some participants felt that rolling for the number of cards they need to use in their pitch proved to be an unpleasant limitation.

> K2: *"I think that having number of cards was kind of hard to do."*
>
> D2: *"Games are very specific on how many cards you get, so when yours was ... seven, ... what would seven be? Three could be resources, four can be turfs ... I was starting to think too much about the specifics."*
>
> K1: *"Maybe the number of cards will come naturally based on the game you make up."*

We should keep these criticisms on the format of the workshop in mind, since they represents frustrations that the designers had that were independent from the concepts at the center of this study - interactive cards and bend gestures.

7.3.4 Evaluating Expressiveness

To summarize this study, I want to reiterate the question: "Are bendable interactive cards expressive?"

I defined expressiveness to be a combination of semantic width and variety and the syntactic width and variety available with the technology. The syntactic elements are in effect the events that I defined in this work. I have strived to expand the width of the syntax in comparison to other bendable devices by incorporating multiple kinds of discrete events, some of which have a contextual significance, as well as the continuous signal. The variety of the syntax is limited, not all of the syntax units can be rearranged to form new "sentences", for example, a bend start event can't follow a peak release event without firing intervening events. However, during this workshop, the low level events (such as bend start and peak release) were never brought up, with designers focusing on the continuous input, mostly in the form of a dynamic sequence, and the higher level events flick and hold bend. It is possible that the short duration of the workshop and the novelty of the interaction scheme were factors in the designers' approach, but overall, the syntactic variety seems low.

The workshop was partially successful at showing the width and variety of the semantic aspect of the concept. Despite some difficulty with the topics, all of the designers could generate ideas for an interactive card system from random criteria with some amount of success, and these ideas were varied for the most part. This was less evident in the third workshop, where ideas were either short on detail or followed a similar bend-untill-you-get-it-right approach that wasn't very card-like. I would say that the interactive card concept, by itself, has promising expressiveness, but the version limited to using bending interaction is far more limited. The conclusion is that multiple sensor would provide more space to design, and the designer would need to choose what interaction works for their game.

CHAPTER 8
CONCLUSION AND FUTURE DIRECTIONS

I called this manuscript *"Research through Design of Bendable Interactive Playing Cards"*, which, in my mind, tied three concepts together in a pretty package with a bow on top. The concepts were 1) Research through Design (RtD), 2) bendable [devices], and 3) Interactive playing cards. I called the "package" I made from these concepts PEPA - Paper-like Entertainment Platform Agents.

At the start of this manuscript, I explained why I find physical/digital hybrid cards interesting, how they combine the benefits of paper card games of social interaction and tactile enjoyment of cards, with the benefits of digital games, that can automate away a lot of fussy parts of the game and have affordances that are hard to achieve with non-digital means. I described several scenarios where interactive playing cards play a role.

I then presented my research questions. I chose 3 specific dimensions of interest to study: suitability, feasibility, and expressivity. I argued that bend gestures are suitable for card games, that functioning bendable interactive cards are feasible to create, and that the resulting system is sufficiently expressive.

I supported my suitability argument with Chapter 3 that includes a meta-analysis of bendable devices and Chapter 4 describing an observational study of players. In my meta-analysis, I tried to point out that despite the research done so far, it is not not clear what bend-gestures might be good for. There was some positive response to the use of bending as a zooming application, but in many other comparisons, bending cannot compete with the now-ubiquitous touch. I also categorized the types of bendable devices in literature and placed my work within the "digital paper" category, which is a promising category that was somewhat neglected with the rise of personal smart devices. I described my observational study to showcase how I've seen players using paper cards perform minute, unintentional bends during significant action points in, what I called, the game's "interaction rhythm". The argument then said: if bends are expected at action points, if bends were also sensed (using sensors for that purpose), and bends were desired by the game - then players could easily translate their experience from paper to digital. Then perhaps bendable cards were the application that bend gestures were looking for.

I approached the feasibility argument by building bendable card devices. When one sits down to build something that they don't know how to do, the process involves questions and decisions they could not foresee in advance. In my case, while I was building prototypes, I faced an obstacle - I was not sure how to implement software for the device, and when I contemplated how to get past this obstacle, I decided I need to understand bend events. In chapter 5 I detailed a user study that helped me develop a model for bend events. In Chapter 6 I went into full details of the design and fabrication of card prototypes. I identified five different versions I created for the prototype, and list for each one a description, goals, details of refinement, and lessons learned. As an aside, I explain my process of reflecting on the deign process using a tool I devised called Designer's Reframe and Refine Diagram (DRRD).

To showcase expressivity, I conducted a design workshop, presented in Chapter 7, where designers were asked to generate pitch ideas for interactive card games. The variety of ideas suggested that interactive cards, as a platform, were very expressive, while the *bendable* interactive cards, limited the expressivity of ideas. Participants commented that they were feeling obligated to come up with ideas that fit bending rather than an interesting card game, and as a result, most ideas had a similar "dynamic sequence" gestural language that took away from the "cardiness" of the games.

Looking back at the concepts I tied together, this dissertation presented a vision for interactive playing cards. It discussed use cases, literature, and an evaluation of expressivity for such cards. For the concept of bendable devices, this dissertation presented a methodical review of devices, theory regarding bend events, an exemplar device, and ideas for a suitable application space. For the RtD concept, I contributed in this manuscript a new tool for reflection and documentation of the RtD process, and provided an example of using this tool with the PEPA project. This last contribution alone can be a catalysis for future research: how will this tool help compare work between different designers, do designers have a style, does designer's experience play a role in the kind of path taken? Can DRRD be used to plan rather than document the design process? Does the distance between reframe versions on the diagram diminish as the project nears its end? However, this is not the focus of this chapter.

My work touched on the subjects of bendable events and interactive cards, and served as an introduction to a larger design space that can fill a complex and rich research agenda. By focusing on suitability, feasibility, and expressivity, I merely scratch the surface of this larger space. In the next sections, I will discuss several possible direction for further research based on, or divergent from, the research in this project. Directions of interest include: developing hardware, networking devices, improving interaction, and exploring application domains.

8.1 Hardware Improvement

The cards I designed in this project were intentionally limited to using off-the-shelf components and processes that are reproducible at a hobbyist home. This decision was based on materials and technology that I had access to, a desire for fast prototyping that would enable me to focus on the interaction, and a preference to make this project openly available to others with minimal technical expectations. They were further limited when I decided to remove the complications resulting from wireless network communication and opted to use a serial connection for both communication and power supply. There are, therefore, many directions that future projects can take in improving (or altogether changing) the current hardware.

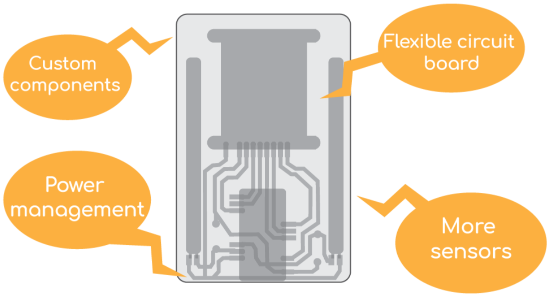


Figure 8.1: Hardware Improvement

First, research into flexible electronic components and boards is currently very active, especially with the promise of such flexible components to support wearable technology [47] and soft robots [154]. Flexible display are already available, there are commercially available e-ink flexible displays and Flexible OLED displays, and researchers look for more ways to make such displays easier to fabricate, for example, using print screening methods [126]. Works explore anything from using conductive ink to draw paper circuits [143] and printing circuits with Inkjet printers [84, 88]. Boem and Troiano shared a review of non-rigid interfaces in HCI research [16] that covers many aspects of flexible electronic components. In this context, advancement of flexible components can promote the vision of interactive cards, making the devices more card-like and more inline with the card-game metaphor.

Second, it was evident in my last study that designers would prefer other input technologies in addition to the bending. This may include novel or off-the-shelf capacitive touch screens, new kinds of touch interactions, such as, back of the device interactions [10] or around the device interaction [99], or a variety of other sensors - accelerometers, IR, pressure, or proximity sensors, to name a few, can be added to detect gestures beyond bends. In addition, new bend sensing technologies can detect more complex bend patterns, FlexSense [135] uses piezoelectric sensors printed in specific patterns to detect curvature of a surface, ShArc sensor [153] can correctly measure multiple bends on the same curve using relative shifts of inner and outer layers of the sensor. Adding higher-level bend sensing as well as other forms of input increases (and changes) the design space of applications for interactive cards.

Third, power management is a key concern: when a system is distributed over multiple devices, it is presumed that all of those devices should be charged and active simultaneously. It is not a likely proposition to connect dozens of card devices to charging cables. There are ubiquitous wireless charging technologies these days (i.e. the Qi charger), however, these may have disadvantages of their own, for example, for most home-based wireless chargers the devices need to be placed directly on the charger and with a specific orientation to work. Other possible ideas, still undeveloped, include charging devices with alternative sources, for example a kinetic harvester gathering energy from human motion [114]. The way in which the system as a whole is charged affects the way it is integrated into the life of users, for example, if it is charged by harvesting kinetic energy, the user will want to carry the deck of cards with them.

While there are many means by which one can redesign the hardware, there is also a need to properly evaluate the devices with a hardware focus. Criteria may include: *lifespan* (how long a card works correctly without crashing, malfunctioning, or breaking), bend *accuracy* (how many cycles of bending back and forth can be performed without significant deterioration is signal reading), *mutability* (how many cycles of bending back and forth can be performed without losing the âĂIJidleâĂİ shape significantly), and *power efficiency* (how long can the battery last, how long does it take it to charge, how fast does it deteriorate).

8.2 Ad-hoc Network

The PEPA system was intended to be a distributed system communicating via wireless network, while that goal shifted during this project, it is still an integral part of the interactive cards concept. Distributed systems are covered in the recent *Cross-Device Taxonomy* by Brudy et al. [19]. This system is positioned between areas of "multi-device environments and spaces" where the boundaries between bespoke devices blur (like Weiser's image of ubiquitous computers [173]) and "ad-hoc, mobile cross-device use" which focuses on flexible setups of devices.

The taxonomy expounds six key dimensions of cross-device design: temporal, configuration, relationship, scale, dynamics, and space. In the case of an interactive card system, the temporal dimension of such a system is *synchronous* (the devices are used simultaneously), the configuration is *logical distribution* (distinct UI elements across multiple devices), relationship covers the full gamut of possibilities from single user to multiple users collaborating, and single device to multiple devices, the scale dimension can be *personal* or *social* (within a person's reach or a group space) but one can also imagine applications for the system in a *public* scale, the dynamics is *ad-hoc, mobile* (card devices can be brought in or taken out of use), and the space dimension is *co-located*. This list of properties helps identify the position of the system in the cross-device design space and previews the difficulties that such a system is likely to face, based on existing work.

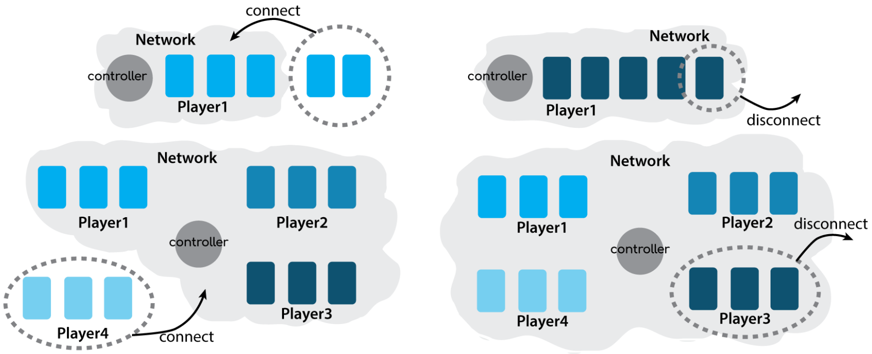


Figure 8.2: Ad-hoc Network

A particularly interesting set of problems stems from the ad-hoc nature of the system. According to the taxonomy, there are three phases of cross-device interaction: 1) Configuration of Devices, 2) Content Engagement, and 3) Disengagement. The configuration phase includes pairing, combining, connecting, or coupling multiple devices and includes various interaction techniques like synchronous tapping [134], identifying proximity [8], and many others [28]. In the case of the PEPA system as an entertainment platform, card devices may need to associate with each other as belonging to a certain player, as well as association as part of a larger system for an active game. The disengagement phase which disconnects connected devices is like-wise complex. Other projects suggested turning away to indicate disconnection [8], tilting a device to indicate its break of connection [109], or providing an explicit application interface for disconnecting devices [62]. Whatever methods used for forming and disengaging the system of cards (as well as content engagement) will depend on the hardware rework for the cards and what kind of interaction techniques they support. Following the configuration and disengagement interaction design, comes a need for visualizing system state to the user - which device is integrated into the cross-device system and in what role. Current solutions to this issue include color coding [62] and using proxy icons [75]. In the PEPA case, association with a specific player may add complexity.

In addition to the many interaction problems that arise from a distributed system, especially an ad-hoc one, there are many technical issues to resolve in the creation of a wireless network: what wireless technology to use? Current implementation suggest BLE (Bluetooth Lite), but other protocols may prove to be more appropriate. Generally, a new device (client) in an ad-hoc network would need to broadcast their identifier data and wait for a central device (server) to form a connection and define the kind of characteristics that will be sent and received. The exact implementation would depend on the wireless technology chosen, which will be wrapped around a PEPA specific protocol.

Some technical criteria for evaluation of the network connectivity may include: *range* (the physical distance cards can be from each other and from their controlling device - pc or tablet), *throughput* (each card is expected to main constant connection with the controlling device and send input values, in addition, every card receives messages from the controlling device containing graphics instructions. The rates of these instructions vary depending on application. Can the network can keep up with the workload? and/or how many cards can be connected at the same time to the system without detrimental effects on the throughput?), *lag* (the delay between sending a message and its execution) *accuracy* (the correct and consistent delivery of messages), and *connectivity detection* (system's ability to detect new connection correctly and offer a graceful fail if the connection is not detected, or a card falls off the network).

8.3 Affordances, Signifiers, and Feedback

In his classic 7 stages of action model, Donald Norman [122] describes the steps that comprise every action: (1) perceiving the state of the world, (2) interpreting that perception, (3) evaluation of interpretations, (4) comparing to goals, (5) intention to act, (6) sequence of action, and (7) executing the action sequence. These are steps an individual would take when trying to complete a task, they describe a cycle where one tries to understand what actions are available to them (evaluation), decide which actions would help them achieve their goals, then form and perform said actions (execution).

This cycle has become second nature to us with idiomatic systems like our computer's mouse-keyboard setup, or our tablet's touch screen. But this was not always so. We had to learn the "language" of using these systems, what are they capable of doing, what visual indicators tell us they are prepared to do, how do these systems interpret our actions, and how do we know our actions were understood. These questions are leads into the concepts of *affordance*, *signifiers*, and *feedback*. The language we learned so far - cursors, icons, change on hover, pinch to zoom, tap and hold - did not magically emerge, it was developed through design and research. To make systems like PEPA viable in the future, a similar language is needed that cover both of its problematic aspects: the distributed nature of the system and the bend interaction.

Affordance is a concept first introduced by J. J. Gibson [51] and popularized in design by Norman [123]. Norman refers to *perceived* affordance as the perception that some action is possible on an object and *real* affordance as the actual properties of the thing. He later noted that rather than over-designing affordances, designers should focus on designing good signifiers [124]. Signifiers include any *"physically perceivable cue"* that communicates to a person what behavior is appropriate. Norman purports that signifiers are the perceivable part of an affordance, and urges designers to provide signifiers [124]. Feedback can be thought of as a form of signifiers that specifically communicates what was actually done to the object after the action.

What does affordance mean on an object that can change it shape? Ishii at el. [78] term "dynamic affordance" as affordance - perceived actions - that change as the interface changes shape. Petersen, Rasmussen, and Trettvik [130] collected their thoughts about affordance of Shape-Changing Interfaces (SCI) - according to the authors, interfaces have both an *affordance* ("real affordance" in this case, things that can be physically done) and *information* which *"is what specifies the world around us, it is available for perception and can be picked up by an active perceiver exploring its surroundings"* (p. 1961). In their examples, affordance does not necessarily change with the interface change - the information does. Neither of these sources comments specifically on adding signifiers or feedback to SCI.

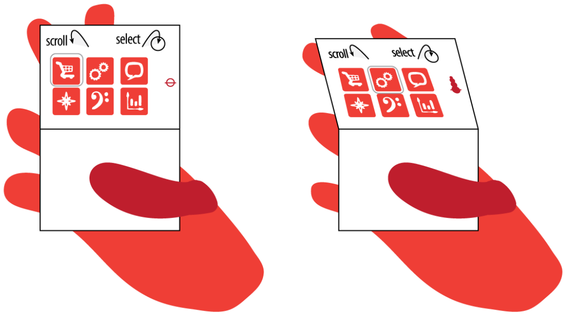


Figure 8.3: Affordances, Signifiers, and Feedback

Within the bendable research community, this topic has not seen many works. The BendyPass device [18] was cast in a silicone mould designed with grooves that create triangles around the areas of the four corners and a vertical groove in its center to provide better affordance for the device - each groove indicates a possible axis for bending. In the paper about PaperNinja [41] which dealt with learnability of bend gestures, the authors provide examples for signifiers for the gestures. For example, to fold the PaperNinja backward two diagonal orange arrows pointing at the top two corners of the device with the text "Down" indicate the desired bend gesture. A similar fold forward action is signified with two blue diagonal arrows pointing away from the top two corners with the text "Up". A "crumple-up-to-a-ball" action is signified by a blue diagonal arrow pointing away from the top-left corner with the text "Up" along with a similar arrow at the bottom-right corner. A work in progress [37] suggest an application of moving carets in a text block using bend gestures, this could be an interesting scenario to test feedback, however, this work has not progressed in that direction.

In the case of the PEPA, the design in this space is complicated not just by the deformable nature of the devices, but also by their distributed state. Some of these issues were mentioned in the previous section (Ad-hoc network). This is a promising line of future research due to the dearth of existing work, and one can imagine many different schemes to signify possibilities or provide feedback, followed up with a comparative usability study.

8.4 Interactive Cards in Display Eco-Systems

At the conception of ubiquitous computing as we know it today, Mark Weiser [173] shared in an essay called "The Computer of the 21s Century" a vision of computers vanishing into the background. He described a system with **tabs**, **pads**, and **boards**, which differ in their sizes (tabs being the size of a note or badge, pads the size of a notepad, and boards the size of a blackboard) and their intended purpose. He imagined that a room may have *"100 tabs, 10-20 pads, and 1-2 boards."* Since then, technology evolved in a different direction than Weiser anticipated - modern "tabs" and "pads" in the form of smart phones and tablets are far more advanced than Weiser could imagine and large displays are often available in large meeting rooms.

These modern tabs, pads, and boards systems can be thought of as *Display Ecologies* [29] (or MDE, Multi-Device Environment) using *Cross Device Interaction* [19]. The sizes of the displays determines the type of interactions a system could support [167]. Large displays are particularly useful for collaborative work and sense-making [4]. Researchers had explored the space of display ecologies that complement large displays with other devices such as personal phones [30, 95], smartwatches [70] and TUIs [45, 158] - these were connected to large displays to serve as either pointers, lenses into a complex piece of data, personal views, UI facilitators, or proxies of elements on the larger display [29].

By varying the sizes and amount of devices in a display ecology, we can explore a large design space of use cases. For example, Horak et al. [70] studied the combination of a large display and personal smartwatche devices in visual analytics tasks. The devices play different roles in this environment: the large display is a public and collaborative space, the smartwatches are used as a user specific storage, mediator, and remote control - the small displays provide a personal point of interaction for each individual performing the task on an otherwise shared resource. In another study, Hamilton et al. [62] used an abundance of homogeneous devices (in this case, tablets) to support text analysis tasks by allowing coupling of information between devices. The authors suggest that spatial arrangement of the physical devices holding the information is instrumental to the analysis task.

In a similar way, the card devices can be likened to the small screens in other device eco-systems. A card device does not have as many features as a smart phone, but this could have benefits: the simpler set of components makes the card cheaper to fabricate and easier to manipulate in an environment that can take advantage of device abundance - perhaps 50 or 100 "tabs", like Weiser's vision. The simpler interaction

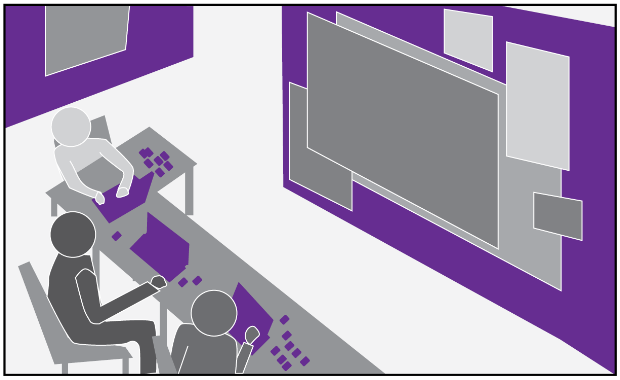


Figure 8.4: Interactive Cards in Display Eco-Systems

scheme can help users focus on the task at hand, rather than get distracted by using multiple applications simultaneously.

For example, a single card can be set to represent 1) a document (possibly to indicate that it is to be reviewed at a later point in time), 2) a clustering of selected documents on the large display and their positions in relation to each other, 3) a snapshot of the large display layout at a point in time that may be of importance later (similar to a commit of a repository which the user may return to at a later point in time). The bend interactions can serve to load, recall, enhance, and dismiss such virtual objects from a card while the user is preforming a task.

8.5 Smarter Event Identification

I have discussed in several parts of this dissertation the analog signal read from the bend sensors in my card devices. Currently, I smooth that signal using a low-pass time filter and use the difference between the current value and the last value, along with the current state in a state machine, to identify a bend event based on the event model I defined. I tried to treat my input device as deterministic, however, it is very likely not the correct approach for an input signal that is noisy to begin within, and an interaction scheme that may be open to interpretation - what is one person's "big" bend compared to another's?

Artificial Intelligence (AI) and Machine Learning (ML) methods can potentially overcome such obstacles, as well as make other, more complex, bending patterns identifiable. For example, Hartmann [64, 63] used pattern recognition to identify a continuous signal pattern generated by an analog sensor. Hartmann's

Exemplar authoring environment uses "programming by demonstration" to create an input pattern and a Dynamic Time Warping (DTW) algorithm to recognize the pattern on new input.

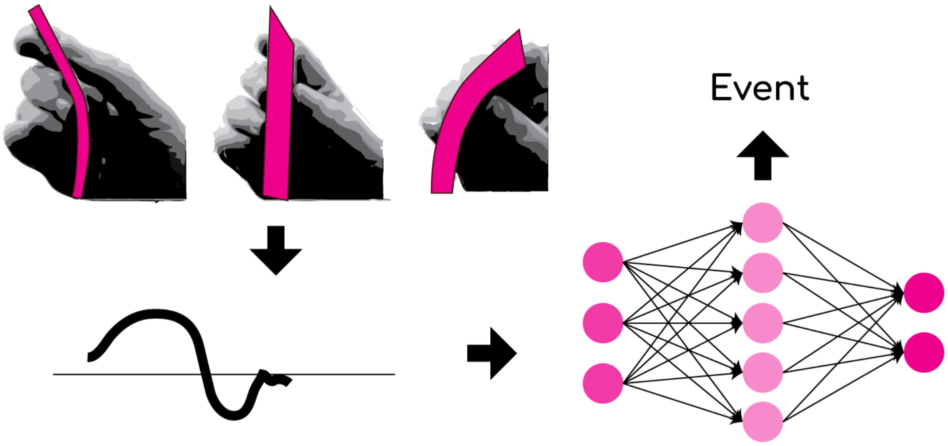


Figure 8.5: Smarter Event Identification

Many modern works in the area of deformable interfaces deal with arrays of sensors, i.e. multiple sensors distributed in a known configuration (often, a tight matrix) where, at any given time, the input from the array is the amalgamation of all signal values across the system. To determine the state of an interface based on multiple simultaneous signals, researchers usually use AI/ML to form extensive models of all the possible states, and recognize specific instances in real-time. FlexSense [135], for example, uses an array of piezoelectric sensors arranged around the circumference and on the corners of a transparent sheet that they use as an overlay on top of existing devices. The paper goes into great lengths to describe a weighted linear interpolation ML algorithm they devised to calculate the deformation of the sheet, as well as how they captured ground truth data to feed their system during training and test times. SmartSleeve [128] suggests various gestures a user can preform on their fabric sleeve (rub, push, stretch, fold etc.) and uses conductive thread to create an array of pressure sensors. The detection algorithm reduces the raw sensor data to a "force image" representing the system state, then uses a trained classifier and heuristics to identify a gesture. Shih et al. [154] present more examples in their review of e-skin sensors for soft robots.

In addition to interpreting gestures and events using sensor signals, there is a substantial body of work on using AI/ML to identify gestures using Image Processing. There are countless works on identifying hand-gestures alone (see Rautaray and Agrawal [133] for a survey) and it is still an active space with designers aspiring to increase accuracy to accommodate Mixed Reality systems (VR/AR) that have built-in cameras as a main form of input. In the field of deformable interfaces, using camera based detection requires a special setup in a room or over-the-shoulder camera. We have already seen a prototype [46] that uses infra-red reflective markers to reconstruct deformation, as well as a more advanced algorithm [161] for detecting deformation of an unmarked sheet of paper. Image processing can also be effective for detecting deformation in non-flat surfaces (i.e. PhotoelasticTouch [146]) making it even more flexible at event identification.

Overall, there are many options worth exploring in this design space: including more sensors and more event "smarts" can increase the type of interactions players can have with the cards and with each other. It would be particularly interesting to design interactive cards that are spatially aware of their location as well as aware of the player holding/looking at them.

8.6 Hybrid Games

In this dissertation, I described the field of enhanced/hybrid games, games that combine digital with physical objects. By their nature, interactive cards (bendable or otherwise) would fall under this category of games. I have also presented a workshop study that prompted designers to come up with pitch ideas for games that are intended for interactive playing cards (bendable or otherwise). However, these pitches, and other game ideas that have come to my mind, remain un-tested. Are these games fun to play? Are the cards easy to use? Does the digital/physical nature of the interactive cards truly contribute something special that cannot be replicated with other technologies? And how do the bend gestures fair, do players enjoy them?

Since this work focused on the basics of the platform, future work should delve deeper into the applications designed for it and their use. From a simple enhancement of existing card games using sound and animation (like the work of Park et al. [127]), to adding game automation and rule management [101], or virtualizing part of a card game to support remote play [145, 117], there are many reasons to implement hybrid games. I started this manuscript by providing six scenarios of usage: automate card play, enrich game experience, guided games, accessibility in playing cards, novel game mechanics, and remote play.

While all of the scenarios provide an avenue of research, the accessibility scenario is particularly interesting. In a recent study, Andrade et al. [3] studied the playing habits "in-the-wild" of visually impaired gamers (GVI). The players relied heavily on audio-based games. The games need to have low complexity, and the player needs to be able to access the "history" of play. They enjoyed the social aspects of play but play mostly with other visual impaired players. With proper design, games can have lower barriers for GVI players, and afford mainstream games to appeal to both sighted and GVI players. It is not clear if bendable interactive cards are suitable for this scenario or if they are more advantageous over other gestural schemes for visually impaired people, however, several researchers have already experimented with bend gestures for blind people [38, 18, 39] and the existence of prior work helps with performing background research when starting a new project.

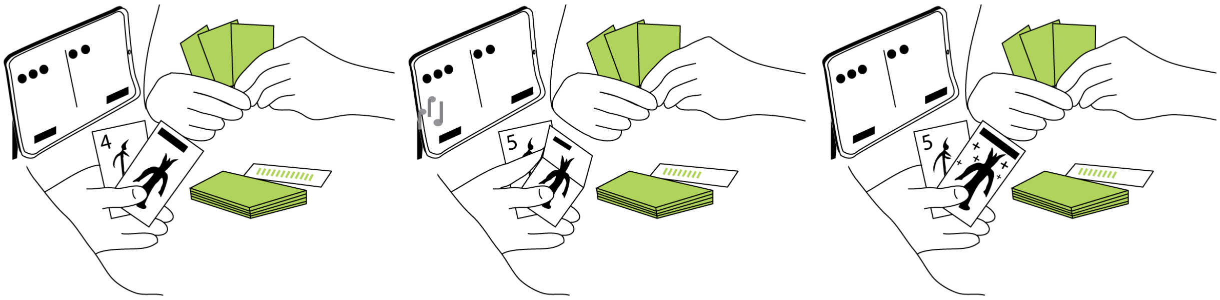


Figure 8.6: Hybrid Games

BIBLIOGRAPHY

[1] S. Adinolf and S. Turkay. Collection, creation and community: a discussion on collectible card games. In *Proceedings of the 7th international conference on Games+ Learning+ Society Conference*, pages 3–11. ETC Press, 2011.

[2] T. T. Ahmaniemi, J. Kildal, and M. Haveri. What is a device bend gesture really good for? In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, CHI '14, pages 3503–3512, New York, NY, USA, 2014. ACM.

[3] R. Andrade, M. J. Rogerson, J. Waycott, S. Baker, and F. Vetere. Playing blind: Revealing the world of gamers with visual impairment. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–14, New York, NY, USA, 2019. ACM.

[4] C. Andrews, A. Endert, and C. North. Space to think: large high-resolution displays for sensemaking. In E. D. Mynatt, D. Schoner, G. Fitzpatrick, S. E. Hudson, W. K. Edwards, and T. Rodden, editors, *Proceedings of the 28th International Conference on Human Factors in Computing Systems, CHI 2010, Atlanta, Georgia, USA, April 10-15, 2010*, pages 55–64. ACM, 2010.

[5] R. Ansara and A. Girouard. Augmenting bend gestures with pressure zones on flexible displays. In *Proceedings of the 16th International Conference on Human-Computer Interaction with Mobile Devices & Services*, MobileHCI '14, pages 531–536, New York, NY, USA, 2014. ACM.

[6] G. Bailey, D. Sahoo, and M. Jones. Paper for e-paper: Towards paper like tangible experience using e-paper. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*, ISS '17, pages 446–449, New York, NY, USA, 2017. ACM.

[7] R. Balakrishnan, G. Fitzmaurice, G. Kurtenbach, and K. Singh. Exploring interactive curve and surface manipulation using a bend and twist sensitive input strip. In *Proceedings of the 1999 Symposium on Interactive 3D Graphics*, I3D '99, pages 111–118, New York, NY, USA, 1999. ACM.

[8] T. Ballendat, N. Marquardt, and S. Greenberg. Proxemic interaction: designing for a proximity and orientation-aware environment. In A. Krüger, J. Schöning, D. Wigdor, and M. Haller, editors, *ACM*

International Conference on Interactive Tabletops and Surfaces, ITS 2010, Saarbrücken, Germany, November 7-10, 2010, pages 121–130. ACM, 2010.

[9] J. Bardzell, S. Bardzell, P. Dalsgaard, S. Gross, and K. Halskov. Documenting the research through design process. In M. Foth, W. Ju, R. Schroeter, and S. Viller, editors, *Proceedings of the 2016 ACM Conference on Designing Interactive Systems, DIS '16, Brisbane, QLD, Australia, June 04 - 08, 2016*, pages 96–107. ACM, 2016.

[10] P. Baudisch and G. Chu. Back-of-device interaction allows creating very small touch devices. In D. R. O. Jr., R. B. Arthur, K. Hinckley, M. R. Morris, S. E. Hudson, and S. Greenberg, editors, *Proceedings of the 27th International Conference on Human Factors in Computing Systems, CHI 2009, Boston, MA, USA, April 4-9, 2009*, pages 1923–1932. ACM, 2009.

[11] V. Bellotti, M. Back, W. K. Edwards, R. E. Grinter, A. Henderson, and C. Lopes. Making sense of sensing systems: five questions for designers and researchers. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 415–422, 2002.

[12] S. Benford, H. Schnädelbach, B. Koleva, R. Anastasi, C. Greenhalgh, T. Rodden, J. Green, A. Ghali, T. Pridmore, B. Gaver, et al. Expected, sensed, and desired: A framework for designing sensing-based interaction. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 12(1):3–30, 2005.

[13] M. Billinghurst, H. Kato, I. Poupyrev, et al. Tangible augmented reality. *ACM SIGGRAPH ASIA*, 7, 2008.

[14] E. Blevis, S. Hauser, and W. Odom. Sharing the hidden treasure in pictorials. *Interactions*, 22(3):32–43, 2015.

[15] I. Blizzard Entertainment. Hearthstone: Heroes of warcraft, 2014.

[16] A. Boem and G. M. Troiano. Non-rigid HCI: A review of deformable interfaces and input. In S. Harrison, S. Bardzell, C. Neustaedter, and D. G. Tatar, editors, *Proceedings of the 2019 on Designing Interactive Systems Conference, DIS 2019, San Diego, CA, USA, June 23-28, 2019*, pages 885–906. ACM, 2019.

[17] J. Bowers. The logic of annotated portfolios: communicating the value of 'research through design'. In *Designing Interactive Systems Conference 2012, DIS '12, Newcastle Upon Tyne, United Kingdom, June 11-15, 2012*, pages 68–77. ACM, 2012.

[18] D. Briotto Faustino and A. Girouard. Bend passwords on bendypass: A user authentication method for people with vision impairment. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '18, pages 435–437, New York, NY, USA, 2018. ACM.

[19] F. Brudy, C. Holz, R. Rädle, C. Wu, S. Houben, C. N. Klokmose, and N. Marquardt. Cross-device taxonomy: Survey, opportunities and challenges of interactions spanning across multiple devices. In S. A. Brewster, G. Fitzpatrick, A. L. Cox, and V. Kostakos, editors, *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI 2019, Glasgow, Scotland, UK, May 04-09, 2019*, page 562. ACM, 2019.

[20] J. Burstyn, A. Banerjee, and R. Vertegaal. Flexview: An evaluation of depth navigation on deformable mobile devices. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*, TEI '13, pages 193–200, New York, NY, USA, 2013. ACM.

[21] J. Burstyn, J. P. Carrascal, and R. Vertegaal. Fitts' law and the effects of input mapping and stiffness on flexible display interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 3649–3658, New York, NY, USA, 2016. ACM.

[22] W. Buxton. Lexical and pragmatic considerations of input structures. *ACM SIGGRAPH Computer Graphics*, 17(1):31–37, 1983.

[23] W. Buxton. Chunking and phrasing and the design of human-computer dialogues (invited paper). In H. Kugler, editor, *Information Processing 86, Proceedings of the IFIP 10th World Computer Congress, Dublin, Ireland, September 1-5, 1986*, pages 475–480. North-Holland/IFIP, 1986.

[24] W. Buxton. A three-state model of graphical input. In *Human-computer interaction-INTERACT*, volume 90, pages 449–456. Citeseer, 1990.

[25] V. P. C and D. Wigdor. Printem: Instant printed circuit boards with standard office printers & inks. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, pages 243–251, New York, NY, USA, 2015. ACM.

[26] S. K. Card, J. D. Mackinlay, and G. G. Robertson. The design space of input devices. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 117–124, 1990.

[27] S. K. Card, J. D. Mackinlay, and G. G. Robertson. A morphological analysis of the design space of input devices. *ACM Transactions on Information Systems (TOIS)*, 9(2):99–122, 1991.

[28] M. K. Chong and H. Gellersen. How groups of users associate wireless devices. In W. E. Mackay, S. A. Brewster, and S. Bødker, editors, *2013 ACM SIGCHI Conference on Human Factors in Computing Systems, CHI '13, Paris, France, April 27 - May 2, 2013*, pages 1559–1568. ACM, 2013.

[29] H. Chung, C. North, S. Joshi, and J. Chen. Four considerations for supporting visual analysis in display ecologies. In M. Chen and G. L. Andrienko, editors, *10th IEEE Conference on Visual Analytics Science and Technology, IEEE VAST 2015, Chicago, IL, USA, October 25-30, 2015*, pages 33–40. IEEE Computer Society, 2015.

[30] H. Chung, C. North, J. Z. Self, S. L. Chu, and F. K. H. Quek. Visporter: facilitating information sharing for collaborative sensemaking on multiple displays. *Pers. Ubiquitous Comput.*, 18(5):1169–1186, 2014.

[31] F. Daliri and A. Girouard. Visual feedforward guides for performing bend gestures on deformable prototypes. In *Graphics Interface*, pages 209–216, 2016.

[32] P. Dalsgaard. Experimental systems in research through design. In J. Kaye, A. Druin, C. Lampe, D. Morris, and J. P. Hourcade, editors, *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, San Jose, CA, USA, May 7-12, 2016*, pages 4991–4996. ACM, 2016.

[33] P. Dalsgaard, K. Halskov, and R. Nielsen. Maps for design reflection. *Artifact*, 2(3-4):176–189, 2008.

[34] P. Dalsgård and K. Halskov. Reflective design documentation. In *Designing Interactive Systems Conference 2012, DIS '12, Newcastle Upon Tyne, United Kingdom, June 11-15, 2012*, pages 428–437. ACM, 2012.

[35] A. Desjardins and C. Key. Parallels, tangents, and loops: Reflections on the'through'part of rtd. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*, pages 2133–2147, 2020.

[36] P. Dourish. Implications for design. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 541–550, 2006.

[37] A. K. Eady and A. Girouard. Caret manipulation using deformable input in mobile devices. In B. Verplank, W. Ju, A. N. Antle, A. Mazalek, and F. F. Mueller, editors, *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction, TEI '15, Stanford, California, USA, January 15-19, 2015*, pages 587–591. ACM, 2015.

[38] M. Ernst and A. Girouard. Bending blindly: Exploring bend gestures for the blind. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '16, pages 2088–2096, New York, NY, USA, 2016. ACM.

[39] M. Ernst and A. Girouard. Exploring haptics for learning bend gestures for the blind. In J. Kaye, A. Druin, C. Lampe, D. Morris, and J. P. Hourcade, editors, *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, San Jose, CA, USA, May 7-12, 2016, Extended Abstracts*, pages 2097–2104. ACM, 2016.

[40] D. Fallman. Why research-oriented design isn't design-oriented research: On the tensions between design and research in an implicit design discipline. *Knowledge, Technology & Policy*, 20(3):193–200, 2007.

[41] E. Fares, V. Cheung, and A. Girouard. Effects of bend gesture training on learnability and memorability in a mobile game. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*, ISS '17, pages 240–245, New York, NY, USA, 2017. ACM.

[42] K. P. Fishkin. A taxonomy for and analysis of tangible interfaces. *Personal and Ubiquitous computing*, 8(5):347–358, 2004.

[43] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology*, 47(6):381, 1954.

[44] M. Flintham, S. Benford, R. Anastasi, T. Hemmings, A. Crabtree, C. Greenhalgh, N. Tandavanitj, M. Adams, and J. Row-Farr. Where on-line meets on the streets: experiences with mobile mixed reality games. In G. Cockton and P. Korhonen, editors, *Proceedings of the 2003 Conference on Human Factors in Computing Systems, CHI 2003, Ft. Lauderdale, Florida, USA, April 5-10, 2003*, pages 569–576. ACM, 2003.

[45] J. Fuchs, R. Rädle, D. Sacha, F. Fischer, and A. Stoffel. Collaborative data analysis with smart tangible devices. In P. C. Wong, D. L. Kao, M. C. Hao, and C. Chen, editors, *Visualization and Data Analysis 2014, San Francisco, CA, USA, February 3-5, 2014*, volume 9017 of *SPIE Proceedings*, page 90170C. SPIE, 2014.

[46] D. T. Gallant, A. G. Seniuk, and R. Vertegaal. Towards more paper-like input: Flexible input devices for foldable interaction styles. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*, UIST '08, pages 283–286, New York, NY, USA, 2008. ACM.

[47] W. Gao, H. Ota, D. Kiriya, K. Takei, and A. Javey. Flexible electronics toward wearable sensing. *Accounts of chemical research*, 52(3):523–533, 2019.

[48] W. Gaver. What should we expect from research through design? In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 937–946, 2012.

[49] W. W. Gaver. Making spaces: how design workbooks work. In D. S. Tan, S. Amershi, B. Begole, W. A. Kellogg, and M. Tungare, editors, *Proceedings of the International Conference on Human Factors in Computing Systems, CHI 2011, Vancouver, BC, Canada, May 7-12, 2011*, pages 1551–1560. ACM, 2011.

[50] E. Ghomi. *Designing expressive interaction techniques for novices inspired by expert activities : the case of musical practice. (La conception de techniques d'interaction expressives adaptées aux novices et inspirées par des activités expertes : le cas de la pratique musicale).* PhD thesis, University of Paris-Sud, Orsay, France, 2012.

[51] J. J. Gibson. The theory of affordances. *Hilldale, USA*, 1(2):67–82, 1977.

[52] A. Girouard, J. Lo, M. Riyadh, F. Daliri, A. K. Eady, and J. Pasquero. One-handed bend interactions with deformable smartphones. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 1509–1518, New York, NY, USA, 2015. ACM.

[53] A. Girouard, A. Tarun, and R. Vertegaal. Displaystacks: Interaction techniques for stacks of flexible thin-film displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2431–2440, New York, NY, USA, 2012. ACM.

[54] O. Glauser, D. Panozzo, O. Hilliges, and O. Sorkine-Hornung. Deformation capture via soft and stretchable sensor arrays. *ACM Trans. Graph.*, 38(2):16:1–16:16, Mar. 2019.

[55] A. Gomes, A. Nesbitt, and R. Vertegaal. Morephone: A study of actuated shape deformations for flexible thin-film smartphone notifications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 583–592, New York, NY, USA, 2013. ACM.

[56] A. Gomes, L. Priyadarshana, J. P. Carrascal, and R. Vertegaal. Whammyphone: Exploring tangible audio manipulation using bend input on a flexible smartphone. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16 Adjunct, pages 159–161, New York, NY, USA, 2016. ACM.

[57] A. Gomes and R. Vertegaal. Paperfold: A shape changing mobile device with multiple reconfigurable electrophoretic magnetic display tiles. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '14, pages 535–538, New York, NY, USA, 2014. ACM.

[58] A. Gomes and R. Vertegaal. Paperfold: Evaluating shape changes for viewport transformations in foldable thin-film display devices. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '15, pages 153–160, New York, NY, USA, 2015. ACM.

[59] M. Green and R. Jacob. Siggraph'90 workshop report: software architectures and metaphors for non-wimp user interfaces. *ACM SIGGRAPH Computer Graphics*, 25(3):229–235, 1991.

[60] S. Greenberg and C. Fitchett. Phidgets: easy development of physical interfaces through physical widgets. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 209–218, 2001.

[61] U. Hagen. Where do game design ideas come from?: Invention and recycling in games developed in sweden, 2009.

[62] P. Hamilton and D. J. Wigdor. Conductor: enabling and understanding cross-device interaction. In M. Jones, P. A. Palanque, A. Schmidt, and T. Grossman, editors, *CHI Conference on Human Factors in Computing Systems, CHI'14, Toronto, ON, Canada - April 26 - May 01, 2014*, pages 2773–2782. ACM, 2014.

[63] B. Hartmann. *Gaining design insight through interaction prototyping tools*. Stanford University Stanford, CA, 2009.

[64] B. Hartmann, L. Abdulla, M. Mittal, and S. R. Klemmer. Authoring sensor-based interactions by demonstration with direct manipulation and pattern recognition. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 145–154, 2007.

[65] G. Herkenrath, T. Karrer, and J. Borchers. Twend: Twisting and bending as new interaction gesture in mobile devices. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '08, pages 3819–3824, New York, NY, USA, 2008. ACM.

[66] K. Hinckley and D. Wigdor. *Input Technologies and Techniques (2012)*, chapter 6, pages 95–132. CRC Press, May 2012.

[67] D. Holman and R. Vertegaal. Organic user interfaces: Designing computers in any way, shape, or form. *Commun. ACM*, 51(6):48–55, June 2008.

[68] D. Holman, R. Vertegaal, M. Altosaar, N. Troje, and D. Johns. Paper windows: Interaction techniques for digital paper. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, pages 591–599, New York, NY, USA, 2005. ACM.

[69] L. E. Holmquist, J. Redström, and P. Ljungstrand. Token-based access to digital information. In *International Symposium on Handheld and Ubiquitous Computing*, pages 234–245. Springer, 1999.

[70] T. Horak, S. K. Badam, N. Elmqvist, and R. Dachselt. When david meets goliath: Combining smartwatches with a large vertical display for visual data exploration. In R. L. Mandryk, M. Hancock, M. Perry, and A. L. Cox, editors, *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, CHI 2018, Montreal, QC, Canada, April 21-26, 2018*, page 19. ACM, 2018.

[71] M. S. Horn and R. J. Jacob. Tangible programming in the classroom with tern. In *CHI'07 extended abstracts on Human factors in computing systems*, pages 1965–1970, 2007.

[72] E. Hornecker. Beyond affordance: tangibles' hybrid nature. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*, pages 175–182, 2012.

[73] E. Hornecker and T. Psik. Using artoolkit markers to build tangible prototypes and simulate other technologies. In *IFIP Conference on Human-Computer Interaction*, pages 30–42. Springer, 2005.

[74] I. Horváth et al. Comparison of three methodological approaches of design research. In *DS 42: Proceedings of ICED 2007, the 16th International Conference on Engineering Design, Paris, France, 28.-31.07. 2007*, pages 361–362, 2007.

[75] S. Houben, P. Tell, and J. E. Bardram. Activityspace: Managing device ecologies in an activity-centric configuration space. In R. Dachselt, T. C. N. Graham, K. Hornbæk, and M. A. Nacenta, editors, *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces, ITS 2014, Dresden, Germany, November 16 - 19, 2014*, pages 119–128. ACM, 2014.

[76] H.-F. Hsieh and S. E. Shannon. Three approaches to qualitative content analysis. *Qualitative health research*, 15(9):1277–1288, 2005.

[77] H. Ishii. The tangible user interface and its evolution. *Communications of the ACM*, 51(6):32–36, 2008.

[78] H. Ishii, D. Lakatos, L. Bonanni, and J. Labrune. Radical atoms: beyond tangible bits, toward transformable materials. *Interactions*, 19(1):38–51, 2012.

[79] H. Ishii, C. Ratti, B. Piper, Y. Wang, A. Biderman, and E. Ben-Joseph. Bringing clay and sand into digital design?continuous tangible user interfaces. *BT technology journal*, 22(4):287–299, 2004.

[80] H. Ishii and B. Ullmer. Tangible bits: Towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '97, pages 234–241, New York, NY, USA, 1997. ACM.

[81] N. Jarvis, D. W. T. Cameron, and A. Boucher. Attention to detail: annotations of a design process. In L. Malmborg and T. Pederson, editors, *Nordic Conference on Human-Computer Interaction, NordiCHI '12, Copenhagen, Denmark, October 14-17, 2012*, pages 11–20. ACM, 2012.

[82] S. Jordà, G. Geiger, M. Alonso, and M. Kaltenbrunner. The reactable: exploring the synergy between live music performance and tabletop tangible interfaces. In *Proceedings of the 1st international conference on Tangible and embedded interaction*, pages 139–146, 2007.

[83] S. Jordà, M. Kaltenbrunner, G. Geiger, and R. Bencina. The reactable. In *ICMC*, 2005.

[84] Y. Kawahara, S. Hodges, B. S. Cook, C. Zhang, and G. D. Abowd. Instant inkjet circuits: lab-based inkjet printing to support rapid prototyping of ubicomp devices. In F. Mattern, S. Santini, J. F. Canny, M. Langheinrich, and J. Rekimoto, editors, *The 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp '13, Zurich, Switzerland, September 8-12, 2013*, pages 363–372. ACM, 2013.

[85] D. V. Keyson and M. Bruns. Empirical research through design. In *Proceedings of the 3rd IASDR Conference on Design Research*, pages 4548–4557, 2009.

[86] M. Khalilbeigi, R. Lissermann, W. Kleine, and J. Steimle. Foldme: Interacting with double-sided foldable displays. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*, TEI '12, pages 33–40, New York, NY, USA, 2012. ACM.

[87] M. Khalilbeigi, R. Lissermann, M. Mühlhäuser, and J. Steimle. Xpaaand: Interaction techniques for rollable displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 2729–2732, New York, NY, USA, 2011. ACM.

[88] A. Khan, J. S. Roo, T. Kraus, and J. Steimle. Soft inkjet circuits: Rapid multi-material fabrication of soft circuits using a commodity inkjet printer. In F. Guimbretière, M. Bernstein, and K. Reinecke, editors, *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology, UIST 2019, New Orleans, LA, USA, October 20-23, 2019*, pages 341–354. ACM, 2019.

[89] J. Kildal. Interacting with deformable user interfaces: effect of material stiffness and type of deformation gesture. In *International Conference on Haptic and Audio Interaction Design*, pages 71–80. Springer, 2012.

[90] J. Kildal, S. Paasovaara, and V. Aaltonen. Kinetic device: Designing interactions with a deformable mobile interface. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '12, pages 1871–1876, New York, NY, USA, 2012. ACM.

[91] J. Kildal and G. Wilson. Feeling it: The roles of stiffness, deformation range and feedback in the control of deformable ui. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction*, ICMI '12, pages 393–400, New York, NY, USA, 2012. ACM.

[92] M. Kipp. Anvil-a generic annotation tool for multimodal dialogue. In *Seventh European Conference on Speech Communication and Technology*, 2001.

[93] N. Kirshenbaum and S. Robertson. Define, refine, and identify events in a bendable interface. In *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces*, ISS '18, pages 345–350, New York, NY, USA, 2018. ACM.

[94] N. Kirshenbaum and S. Robertson. Pepa deck: Bringing interactivity to playing cards. In *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*, CHI PLAY '18 Extended Abstracts, pages 479–486, New York, NY, USA, 2018. ACM.

[95] U. Kister, K. Klamka, C. Tominski, and R. Dachselt. Grasp: Combining spatially-aware mobile devices and a display wall for graph visualization and interaction. *Comput. Graph. Forum*, 36(3):503–514, 2017.

[96] S. R. Klemmer and J. A. Landay. Toolkit support for integrating physical and digital interactions. *Human–Computer Interaction*, 24(3):315–366, 2009.

[97] S. R. Klemmer, J. Li, J. Lin, and J. A. Landay. Papier-mache: toolkit support for tangible input. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 399–406, 2004.

[98] M. Kosa and P. Spronck. What tabletop players think about augmented tabletop games: a content analysis. In S. Dahlskog, S. Deterding, J. M. Font, M. Khandaker, C. M. Olsson, S. Risi, and C. Salge, editors, *Proceedings of the 13th International Conference on the Foundations of Digital Games, FDG 2018, Malmö, Sweden, August 07-10, 2018*, pages 6:1–6:8. ACM, 2018.

[99] S. G. Kratz and M. Rohs. Hoverflow: expanding the design space of around-device interaction. In R. Oppermann, M. Eisenhauer, M. Jarke, and V. Wulf, editors, *Proceedings of the 11th Conference on Human-Computer Interaction with Mobile Devices and Services, Mobile HCI 2009, Bonn, Germany, September 15-18, 2009*, ACM International Conference Proceeding Series. ACM, 2009.

[100] B. Lahey, A. Girouard, W. Burleson, and R. Vertegaal. Paperphone: Understanding the use of bend gestures in mobile devices with flexible electronic paper displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 1303–1312, New York, NY, USA, 2011. ACM.

[101] A. H. Lam, K. C. Chow, E. H. Yau, and M. R. Lyu. Art: augmented reality table for interactive trading card game. In *Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications*, pages 357–360. ACM, 2006.

[102] S.-S. Lee, S. Kim, B. Jin, E. Choi, B. Kim, X. Jia, D. Kim, and K.-p. Lee. How users manipulate deformable displays as input devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 1647–1656, New York, NY, USA, 2010. ACM.

[103] S.-s. Lee, Y.-k. Lim, and K.-P. Lee. Exploring the effects of size on deformable user interfaces. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services Companion*, MobileHCI '12, pages 89–94, New York, NY, USA, 2012. ACM.

[104] D. Leithinger, S. Follmer, A. Olwal, and H. Ishii. Shape displays: Spatial interaction with dynamic physical form. *IEEE computer graphics and applications*, 35(5):5–11, 2015.

[105] J. Lepinski and R. Vertegaal. Cloth displays: Interacting with drapable textile screens. In *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '11, pages 285–288, New York, NY, USA, 2011. ACM.

[106] J. Lo and A. Girouard. Fabricating bendy: Design and development of deformable prototypes. *IEEE Pervasive Computing*, 13(3):40–46, 2014.

[107] J. Lo and A. Girouard. Bendy: Exploring mobile gaming with flexible devices. In *Proceedings of the Eleventh International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '17, pages 163–172, New York, NY, USA, 2017. ACM.

[108] J. Löwgren. On the significance of making in interaction design research. *interactions*, 23(3):26–33, 2016.

[109] A. Lucero, J. Holopainen, and T. Jokela. Pass-them-around: collaborative use of mobile phones for photo sharing. In D. S. Tan, S. Amershi, B. Begole, W. A. Kellogg, and M. Tungare, editors, *Proceedings of the International Conference on Human Factors in Computing Systems, CHI 2011, Vancouver, BC, Canada, May 7-12, 2011*, pages 1787–1796. ACM, 2011.

[110] S. Lundgren and S. Björk. Game mechanics: Describing computer-augmented games in terms of interaction. In *Proceedings of TIDSE*, volume 3, 2003.

[111] I. S. MacKenzie. A note on the information-theoretic basis for fitts' law. *Journal of motor behavior*, 21(3):323–330, 1989.

[112] C. Magerkurth, A. D. Cheok, R. L. Mandryk, and T. Nilsen. Pervasive games: bringing computer entertainment back to the real world. *Comput. Entertain.*, 3(3):4, 2005.

[113] C. Magerkurth, T. Engelke, and D. Grollman. A component based architecture for distributed, pervasive gaming applications. In *Proceedings of the 2006 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, ACE '06, New York, NY, USA, 2006. ACM.

[114] M. Magno, L. Spadaro, J. Singh, and L. Benini. Kinetic energy harvesting: Toward autonomous wearable sensing for internet of things. In *2016 International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM)*, pages 248–254. IEEE, 2016.

[115] R. L. Mandryk and D. S. Maranan. False prophets: exploring hybrid board/video games. In L. G. Terveen and D. R. Wixon, editors, *Extended abstracts of the 2002 Conference on Human Factors in Computing Systems, CHI 2002, Minneapolis, Minnesota, USA, April 20-25, 2002*, pages 640–641. ACM, 2002.

[116] S. Maqsood, S. Chiasson, and A. Girouard. Bend passwords: Using gestures to authenticate on flexible devices. *Personal Ubiquitous Comput.*, 20(4):573–600, Aug. 2016.

[117] B. Maurer and V. Fuchsberger. Dislocated boardgames: Design potentials for remote tangible play. *Multimodal technologies and interaction*, 3(4):72–, 2019.

[118] K. McGoldrick. Mobile friendly rollable displays. In *2006 European Solid-State Device Research Conference*, pages 1–2, Sep. 2006.

[119] D. J. Merrill. *Interaction with embodied media*. PhD thesis, Massachusetts Institute of Technology, 2009.

[120] Y. Nakagawa, A. Kamimura, and Y. Kawaguchi. Mimictile: A variable stiffness deformable user interface for mobile devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 745–748, New York, NY, USA, 2012. ACM.

[121] V. P. Nguyen, S. H. Yoon, A. Verma, and K. Ramani. Bendid: Flexible interface for localized deformation recognition. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '14, pages 553–557, New York, NY, USA, 2014. ACM.

[122] D. A. Norman. *User centered system design: New perspectives on human-computer interaction*. CRC Press, 1986.

[123] D. A. Norman. *The psychology of everyday things.* Basic books, 1988.

[124] D. A. Norman. The way I see it - signifiers, not affordances. *Interactions*, 15(6):18–19, 2008.

[125] W. of the Coast. Magic: The gathering, 1995.

[126] S. Olberding, M. Wessely, and J. Steimle. Printscreen: Fabricating highly customizable thin-film touch-displays. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pages 281–290, New York, NY, USA, 2014. ACM.

[127] W. V. Park, J. H. Jang, S. W. Kang, C. G. Song, S. J. Kim, J. Kim, and Y. W. Ko. Trading card game exploiting rfid and 3d graphic. In *Human-Centric Computing (HumanCom), 2010 3rd International Conference on*, pages 1–5. IEEE, 2010.

[128] P. Parzer, A. Sharma, A. Vogl, J. Steimle, A. Olwal, and M. Haller. Smartsleeve: Real-time sensing of surface and deformation gestures on flexible, interactive textiles, using a hybrid gesture detection pipeline. In K. Gajos, J. Mankoff, and C. Harrison, editors, *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology, UIST 2017, Quebec City, QC, Canada, October 22 - 25, 2017*, pages 565–577. ACM, 2017.

[129] J. Peitz, S. Björk, and A. Jäppinen. Wizard's apprentice gameplay-oriented design of a computer-augmented board game. In H. Ishii, N. Lee, S. Natkin, and K. Tsushima, editors, *Proceedings of the International Conference on Advances in Computer Entertainment Technology, ACE 2006, Hollywood, California, USA, June 14-16, 2006*, page 18. ACM, 2006.

[130] M. G. Petersen, M. K. Rasmussen, and J. Trettvik. Affordances of shape-changing interfaces: An information perspective on transformability and movement. In R. Wakkary, K. Andersen, W. Odom, A. Desjardins, and M. G. Petersen, editors, *DIS '20: Designing Interactive Systems Conference 2020, Eindhoven, The Netherlands, July 6-10, 2020*, pages 1959–1971. ACM, 2020.

[131] J. Pierce. On the presentation and production of design research artifacts in hci. In *Proceedings of the 2014 conference on Designing interactive systems*, pages 735–744, 2014.

[132] R. Ramakers, J. Schöning, and K. Luyten. Paddle: Highly deformable mobile devices with physical controls. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 2569–2578, New York, NY, USA, 2014. ACM.

[133] S. S. Rautaray and A. Agrawal. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review*, 43(1):1–54, 2015.

[134] J. Rekimoto. Synctap: synchronous user operation for spontaneous network connection. *Pers. Ubiquitous Comput.*, 8(2):126–134, 2004.

[135] C. Rendl, D. Kim, S. Fanello, P. Parzer, C. Rhemann, J. Taylor, M. Zirkl, G. Scheipl, T. Rothländer, M. Haller, and S. Izadi. Flexsense: A transparent self-sensing deformable surface. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pages 129–138, New York, NY, USA, 2014. ACM.

[136] C. Rendl, D. Kim, P. Parzer, S. Fanello, M. Zirkl, G. Scheipl, M. Haller, and S. Izadi. Flexcase: Enhancing mobile interaction with a flexible sensing and display cover. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 5138–5150, New York, NY, USA, 2016. ACM.

[137] Y. Rogers and H. Muller. A framework for designing sensor-based interactions to promote exploration and reflection in play. *International Journal of Human-Computer Studies*, 64(1):1–14, 2006.

[138] M. J. Rogerson, M. R. Gibbs, and W. Smith. Digitising boardgames: Issues and tensions. In S. Björk, A. Waern, and M. Fuchs, editors, *Proceedings of the 2015 DiGRA International Conference: Diversity of Play, DiGRA 2015, Lüneburg, Germany, May 14-17, 2015*. Digital Games Research Association, 2015.

[139] M. J. Rogerson, M. R. Gibbs, and W. Smith. "i love all the bits": The materiality of boardgames. In J. Kaye, A. Druin, C. Lampe, D. Morris, and J. P. Hourcade, editors, *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, San Jose, CA, USA, May 7-12, 2016*, pages 3956–3969. ACM, 2016.

[140] M. J. Rogerson, L. A. Sparrow, and M. R. Gibbs. Unpacking "boardgames with apps": The hybrid digital boardgame model. In Y. Kitamura, A. Quigley, K. Isbister, T. Igarashi, P. Bjørn, and S. M. Drucker, editors, *CHI '21: CHI Conference on Human Factors in Computing Systems, Virtual Event / Yokohama, Japan, May 8-13, 2021*, pages 111:1–111:17. ACM, 2021.

[141] K. Römer and S. Domnitcheva. Smart playing cards: A ubiquitous computing game. *Personal and Ubiquitous Computing*, 6(5-6):371–377, 2002.

[142] M. Rooke and R. Vertegaal. Physics on display: Tangible graphics on hexagonal bezel-less screens. In *Proceedings of the Fourth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '10, pages 233–236, New York, NY, USA, 2010. ACM.

[143] A. Russo, B. Y. Ahn, J. J. Adams, E. B. Duoss, J. T. Bernhard, and J. A. Lewis. Pen-on-paper flexible electronics. *Advanced materials*, 23(30):3426–3430, 2011.

[144] M. Sakamoto, T. Alexandrova, and T. Nakajima. Analyzing the effects of virtualizing and augmenting trading card game based on the player's personality. In *Proceedings of The Sixth International Conference on Advances in Computer-Human Interactions*, pages 348–357. Citeseer, 2013.

[145] M. Sakamoto, T. Alexandrova, and T. Nakajima. Augmenting remote trading card play with virtual characters used in animation and game stories. In *Proceedings of the 6th International Conference on ACHI*, volume 2013, 2013.

[146] T. Sato, H. Mamiya, H. Koike, and K. Fukuchi. Photoelastictouch: transparent rubbery tangible interface using an LCD and photoelasticity. In A. D. Wilson and F. Guimbretière, editors, *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology, Victoria, BC, Canada, October 4-7, 2009*, pages 43–50. ACM, 2009.

[147] S. Savic and J. Huang. Research through design: What does it mean for a design artefact to be developed in the scientific context?, 2014.

[148] J. Scholtz and S. Consolvo. Toward a framework for evaluating ubiquitous computing applications. *IEEE Pervasive Computing*, 3(2):82–88, 2004.

[149] M. Schreier. *Qualitative content analysis in practice*. SAGE, Los Angeles, 2012 - 2012.

[150] C. Schwesig, I. Poupyrev, and E. Mori. Gummi: A bendable computer. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 263–270, New York, NY, USA, 2004. ACM.

[151] O. Shaer and E. Hornecker. Tangible user interfaces: Past, present, and future directions. *Found. Trends Hum.-Comput. Interact.*, 3(1–2):1–137, Jan. 2010.

[152] O. Shaer, N. Leland, E. H. Calvillo-Gamez, and R. J. Jacob. The tac paradigm: specifying tangible user interfaces. *Personal and Ubiquitous Computing*, 8(5):359–369, 2004.

[153] F. Shahmiri and P. H. Dietz. Sharc: A geometric technique for multi-bend/shape sensing. In R. Bernhaupt, F. F. Mueller, D. Verweij, J. Andres, J. McGrenere, A. Cockburn, I. Avellino, A. Goguey, P. Bjøn, S. Zhao, B. P. Samson, and R. Kocielnik, editors, *CHI '20: CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, April 25-30, 2020*, pages 1–12. ACM, 2020.

[154] B. Shih, D. S. Shah, J. Li, T. G. Thuruthel, Y. Park, F. Iida, Z. Bao, R. Kramer-Bottiglio, and M. T. Tolley. Electronic skins and machine learning for intelligent soft robots. *Sci. Robotics*, 5(41):9239, 2020.

[155] B. Shneiderman. 1.1 direct manipulation: a step beyond programming languages. *Sparks of innovation in human-computer interaction*, 17:13, 1983.

[156] P. Shorey and A. Girouard. Bendtroller:: An exploration of in-game action mappings with a deformable game controller. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 1447–1458, New York, NY, USA, 2017. ACM.

[157] I. Sony. Eye of judgment, 2007.

[158] M. Spindler, C. Tominski, H. Schumann, and R. Dachselt. Tangible views for information visualization. In A. Krüger, J. Schöning, D. Wigdor, and M. Haller, editors, *ACM International Conference on Interactive Tabletops and Surfaces, ITS 2010, Saarbrücken, Germany, November 7-10, 2010*, pages 157–166. ACM, 2010.

[159] P. Stappers and E. Giaccardi. Research through design. *The encyclopedia of human-computer interaction*, 2, 2017.

[160] T. Starner, B. Leibe, B. Singletary, and J. Pair. MIND-WARPING: towards creating a compelling collaborative augmented reality game. In D. Riecken, D. Benyon, and H. Lieberman, editors, *Pro-*

ceedings of the 5th International Conference on Intelligent User Interfaces, IUI 2000, New Orleans, LA, USA, January 9-12, 2000, pages 256–259. ACM, 2000.

[161] J. Steimle, A. Jordt, and P. Maes. Flexpad: Highly flexible bending interactions for projected handheld displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 237–246, New York, NY, USA, 2013. ACM.

[162] E. Stolterman. The nature of design practice and implications for interaction design research. *International Journal of Design*, 2(1), 2008.

[163] P. Strohmeier, J. Burstyn, J. P. Carrascal, V. Levesque, and R. Vertegaal. Reflex: A flexible smartphone with active haptic feedback for bend input. In *Proceedings of the TEI '16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '16, pages 185–192, New York, NY, USA, 2016. ACM.

[164] M. Sturdee and J. Alexander. Analysis and classification of shape-changing interfaces for design and application-based research. *ACM Comput. Surv.*, 51(1):2:1–2:32, Jan. 2018.

[165] T. Tajika, T. Yonezawa, and N. Mitsunaga. Intuitive page-turning interface of e-books on flexible e-paper based on user studies. In *Proceedings of the 16th ACM International Conference on Multimedia*, MM '08, pages 793–796, New York, NY, USA, 2008. ACM.

[166] A. P. Tarun, P. Wang, A. Girouard, P. Strohmeier, D. Reilly, and R. Vertegaal. Papertab: An electronic paper computer with multiple large flexible electrophoretic displays. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, pages 3131–3134, New York, NY, USA, 2013. ACM.

[167] L. Terrenghi, A. J. Quigley, and A. J. Dix. A taxonomy for and analysis of multi-person-display ecosystems. *Pers. Ubiquitous Comput.*, 13(8):583–598, 2009.

[168] Venturebeat. "superdata: Hearthstone trumps all comers in card market that will hit 1.4 billion dollars in 2017", January 2017. Retrieved Febuary 22, 2018 from `https://venturebeat.com/2017/01/28/superdata-hearthstone-trumps-all-comers-in-card-market-that-will-hit-1-4-billion-in-2017`

[169] J. R. Wallace, J. A. Pape, Y. B. Chang, P. J. McClelland, T. C. N. Graham, S. D. Scott, and M. S. Hancock. Exploring automation in digital tabletop board game. In S. E. Poltrock, C. Simone, J. Grudin, G. Mark, and J. Riedl, editors, *CSCW '12 Computer Supported Cooperative Work, Seattle, WA, USA, February 11-15, 2012 - Companion Volume*, pages 231–234. ACM, 2012.

[170] D. Wang, C. Zhang, and H. Wang. T-maze: a tangible programming tool for children. In *Proceedings of the 10th international conference on interaction design and children*, pages 127–135, 2011.

[171] K. Warren, J. Lo, V. Vadgama, and A. Girouard. Bending the rules: Bend gesture classification for flexible displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 607–610, New York, NY, USA, 2013. ACM.

[172] J.-i. Watanabe, A. Mochizuki, and Y. Horry. Bookisheet: Bendable device for browsing content using the metaphor of leafing through the pages. In *Proceedings of the 10th International Conference on Ubiquitous Computing*, UbiComp '08, pages 360–369, New York, NY, USA, 2008. ACM.

[173] M. Weiser. The computer for the 21st century. *Scientific american*, 265(3):94–105, 1991.

[174] M. Weiser. Some computer science issues in ubiquitous computing. *Communications of the ACM*, 36(7):75–84, 1993.

[175] A. Wiethoff, H. Schneider, J. Küfner, M. Rohs, A. Butz, and S. Greenberg. Paperbox: a toolkit for exploring tangible interaction on interactive surfaces. In *Proceedings of the 9th ACM Conference on Creativity & Cognition*, pages 64–73, 2013.

[176] A. Wiethoff, H. Schneider, M. Rohs, A. Butz, and S. Greenberg. Sketch-a-tui: low cost prototyping of tangible interactions using cardboard and conductive ink. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*, pages 309–312, 2012.

[177] D. Wightman, T. Ginn, and R. Vertegaal. Bendflip: examining input techniques for electronic book readers with flexible form factors. In *IFIP Conference on Human-Computer Interaction*, pages 117–133. Springer, 2011.

[178] A. D. Wilson. Sensor-and recognition-based input for interaction. *The Human Computer Interaction Handbook*, pages 177–200, 2007.

[179] T. V. Wolf, J. A. Rode, J. B. Sussman, and W. A. Kellogg. Dispelling "design" as the black art of CHI. In R. E. Grinter, T. Rodden, P. M. Aoki, E. Cutrell, R. Jeffries, and G. M. Olson, editors, *Proceedings of the 2006 Conference on Human Factors in Computing Systems, CHI 2006, Montréal, Québec, Canada, April 22-27, 2006*, pages 521–530. ACM, 2006.

[180] Y. Xu, E. Barba, I. Radu, M. Gandy, and B. MacIntyre. Chores are fun: Understanding social play in board games for digital tabletop game design. In M. Copier, A. Waern, and H. W. Kennedy, editors, *Proceedings of the 2011 DiGRA International Conference: Think, Design, Play, DiGRA 2011, Hilversum, The Netherlands, September 14-17, 2011*. Digital Games Research Association, 2011.

[181] Z. Ye and H. Khalid. Cobra: Flexible displays for mobilegaming scenarios. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '10, pages 4363–4368, New York, NY, USA, 2010. ACM.

[182] J. Zimmerman, J. Forlizzi, and S. Evenson. Research through design as a method for interaction design research in hci. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 493–502, 2007.

[183] T. G. Zimmerman, J. Lanier, C. Blanchard, S. Bryson, and Y. Harvill. A hand gesture interface device. In *ACM SIGCHI Bulletin*, volume 18, pages 189–192. ACM, 1987.